User's Manual
TK-78K0R/KE3L

Date published: November 2008
Rev. 1.0

Printed in Japan

CAUTION

- Do not give any physical damage to this equipment such as dropping
- Do not superimpose voltage to this equipment.
- Do not use this equipment with the temperature below 0℃ or over 40℃.
- Make sure the USB cables are properly connected.
- Do not bend or stretch the USB cables.
- Keep this equipment away from water.
- Take extra care to electric shock.
- This equipment should be handled like a CMOS semiconductor device. The user must take all precautions to avoid build-up of static electricity while working with this equipment.
- All test and measurement tool including the workbench must be grounded.
- The user/operator must be grounded using the wrist strap.
- The connectors and/or device pins should not be touched with bare hands.

Contents

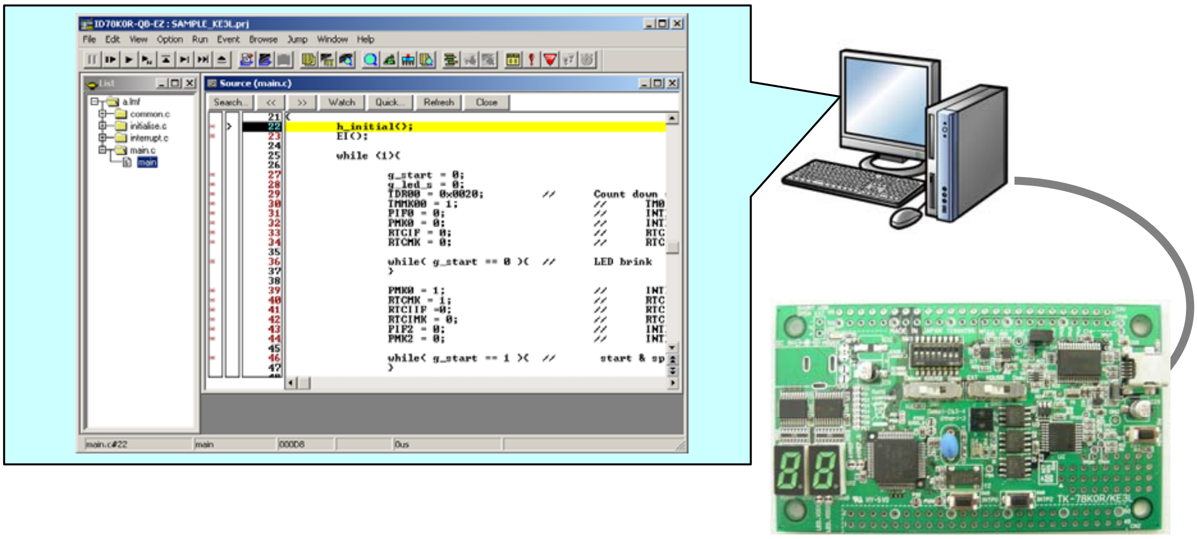
CHAPTER 1 PREPARATION....7
1.1 Development Tools / Software....8
1.2 Installation of Development Tools....9
1.2.1 Installation Package....9
1.2.2 Installation of Development Tools....9
1.3 Installation of USB Driver....14
1.3.1 Installation on Windows XP....15
1.3.2 Installation on Windows 2000....18
1.3.3 Completion of USB Driver Installation....21
1.4 Sample Programs....22
1.4.1 Preparation of Sample Programs....22
CHAPTER 2 EXPERIENCES....24
2.1 Start PM+....26
2.2 What is PM+....27
2.3 Load Workspace (project)....29
2.4 Set Linker Options....31
2.4.1 "Output1" Tab....31
2.4.2 "Output2" Tab....33
2.5 Set Compiler Options....34
2.5.1 "Extend" Tab....34
2.5.2 "Startup Routine" Tab....35
2.6 Create Load Module Files....36
2.7 Check Debugger Settings....38
2.8 Check Board Settings....40
2.9 Start Debugger (ID78K0R-QB-EZ)....41
2.10 Run Programs....45
2.11 Stop Programs....47
2.12 Close Debugger (ID78K0R-QB-EZ)....48
2.13 Quit PM+....49
CHAPTER 3 HARDWARE SPECIFICATIONS....50
3.1 Layout of hardware functions....51
3.2 Hardware Functions....52
3.2.1 SW1....52
3.2.2 SW2....52
3.2.3 SW3....52
3.2.4 SW4 (RESET SW)....53
3.2.5 SW5 (INTP1)....53
3.2.6 SW6 (INTP2)....53
3.2.7 JP1....53
3.2.8 JP2....53

3.2.9 Q4 (Illuminance Sensor) 54
3.2.10 U10, U12 (7segLED) 54
3.2.11 LED1 (POWER) 55
3.3 Universal Area 55
CHAPTER 4 TROUBLESHOOTING 56
4.1 If you cannot find USB driver when you connect PC to the kit 56
4.2 Error when you start the debugger 56
4.2.1 "Can not communicate with Emulator..." (F0100 or A0109) 57
4.2.2 "Incorrect ID Code." (Ff603) 58
4.2.3 "The on-chip debug function had been disabled in the device." (F0c79) 58
4.2.4 "Disabling the on-chip debug function is prohibited." (F0c33) 58
CHAPTER 5 OTHER INFORMATION 59
5.1 Create a new workspace 60
5.2 Register additional source file 65
5.3 Debugger tips 67
5.3.1 Change display of buttons 67
5.3.2 Display source list and function list 67
5.3.3 Set/delete breakpoints 68
5.3.4 Display global variables 69
5.3.5 Display global variables while programs are running 70
5.3.6 Display local variables 71
5.3.7 Display memory and SFR contents 71
5.4 Erase microcontroller built-in flash memory 72
5.5 Circuit diagram 76

Introduction

TK-78K0R/KE3L is the evaluation kit for development with sound systems using "78K0R/Kx3-L", NEC Electronics 16bit all flash microcontroller.
The user only needs to install the development tools and USB driver, and connect the host machine with the target board to start the code development, build, monitoring the output, and debugging code.
(This demonstration kit uses the on-chip debug feature from the microcontroller itself, without emulator connection)

Configuration for Debugging



The user can check the low power consumption of 78K0R/Kx3-L instantly, since the low power consumption demonstration program is written on the system. For details, refer to the user's manual "TK-78K0R/KE3L Low Power Consumption Demonstration GUI".

Overview This manual consists of the following contents.

Read chapter 1 and 2 first for installing the development tools and using the sample programs.

Read chapter 3-5 for customizing the sample programs and the hardware.

Chapter 1: Preparations

Install the development tools

Chapter 2: Experiences

Experience the basic operations of integrated development environment (PM+) and integrated debugger (ID78K0R-QB-EZ) with using sample programs.

Chapter 3: Hardware Specifications

Explain the hardware of TK-78K0R/KE3L

Chapter 4: Troubleshooting

Describe how to solve troubles you may face, such as errors when starting the integrated debugger (ID78K0R-QB-EZ)

Chapter 5: Other Information

Introduce other information, such as how to create a new workspace (project) on integrated development environment (PM+), how to register additional source file, and some useful tips of the integrated debugger. The circuit diagrams of demonstration kit are included in this chapter.

Reader This manual is intended for development engineers who wish to become familiar with the development tools for the 78K0R.

It is assumed that the readers have been familiar with basics of microcontrollers, C and Assembler languages, and the Windows™ operating system.

Purpose This manual is intended to give users an understanding of the features, hardware configurations, development tools for the 78K0R.

CHAPTER 1　Preparation

This chapter describes following topics:

- Overview and installation of development tools
- Installation of development tools
- Overview and preparation of sample programs

Users can experience the development flow such as coding, build, debugging, and test, by using the development tools bundled with TK-78K0R/KE3L.

1.1 Development Tools / Software

- Device file DF781009 V1.10

  A device file contains device specific information. So, users need a device file to use the development tools.

- Integrated Development Environment (IDE) PM+ V6.31

  The IDE works on Windows operation system.
  Users can develop a system efficiently by using the editor with idea processor function, compiler, and debugger.

- C Compiler CC78K0R W2.10 (code size limited version)

  C compiler for the 78K0R microcontrollers. The object code size is limited to 64 Kbyte.
  This compiles C code for 78K0R and ANSI-C code program into assembler code.
  This produces object code and linker.

- Assembler RA78K0R W1.31 (code size limited version)

  Assembler for the 78K0R microcontrollers. The object code size is limited to 64 Kbyte.
  This convert the assembler code for 78K0R into object program. The object program will be used for debugger.

- 78K0R Integrated Debugger ID78K0R-QB-EZ V3.50

  This is the tool for debugging the object program generated by C compiler and assembler. The debugger enables to do C source level debugging. With the debugger, you can debug the code easily and efficiently by refering and changing variables, using step-in debuging function, and so on.

- WriteEZ4

  This is the tool to write HEX file on microcontroller built-in memory without using the debugger (ID78K0R-QB-EZ).

- Starter Kit USB Driver

  USB driver for the USB connection with the TK-78K0R/KE3L and PC.

- Sample program

  Sample program to check the development tools.

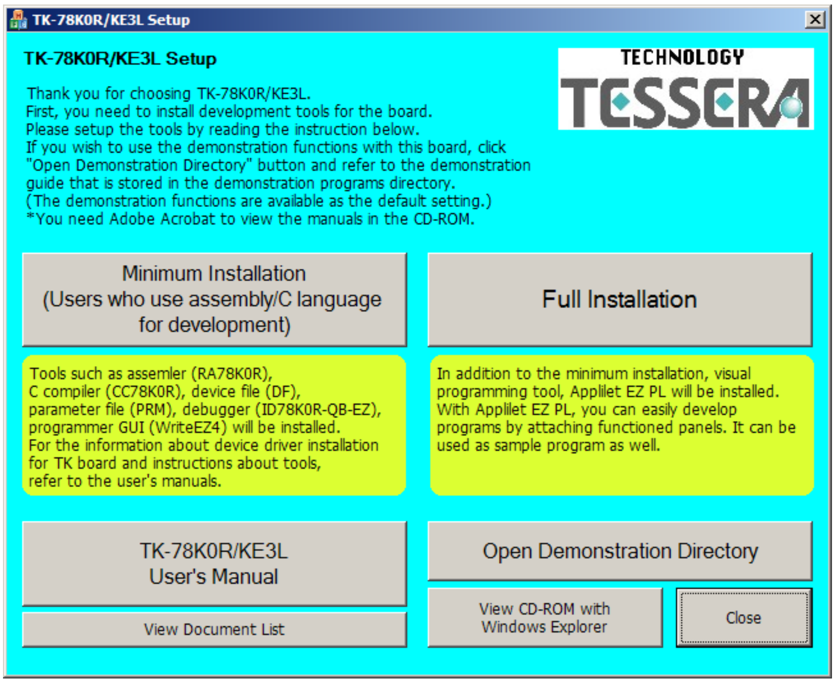
1.2 Installation of Development Tools

1.2.1 Installation Package

The attached CD-ROM includes the development tools and documentations. Users can use the installer to install those development tools and documentations.

1.2.2 Installation of Development Tools

① Please insert the CD-ROM in the drive. The installer will show up automatically.
If it does not start automatically, please initiate it by double clicking the SETUP.EXE.

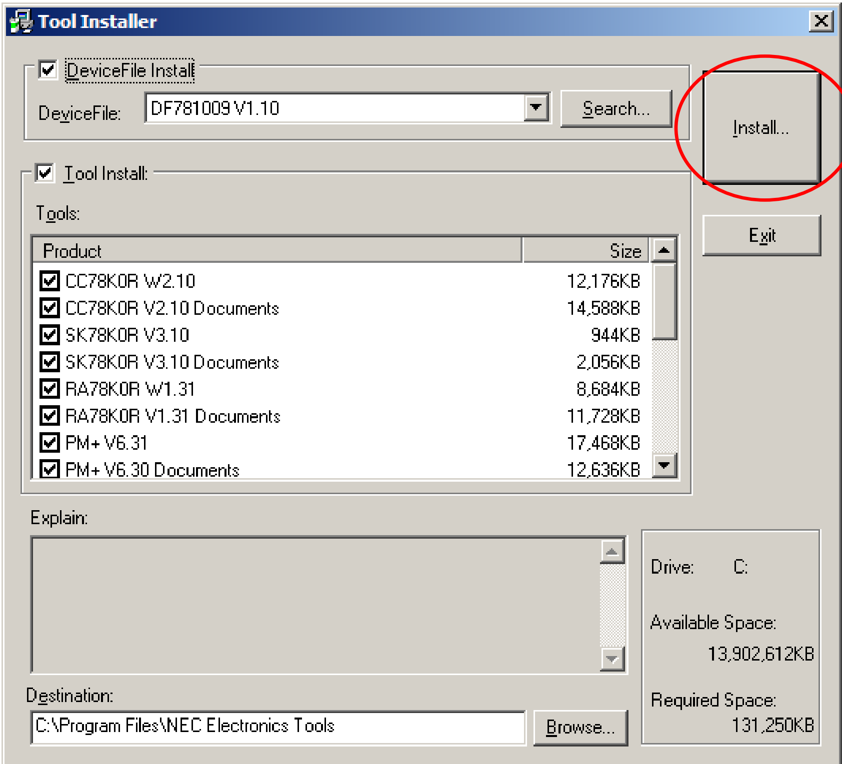


② Click [Minimum Installation] or [Full Installation] button.

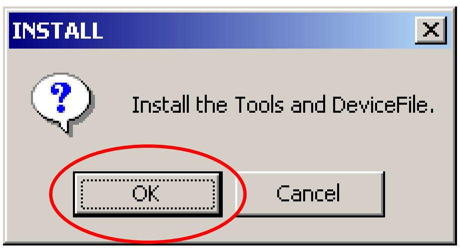
③ "Tool Installer" dialog box is opened.
Select products that you need to install.
(as default, all the products that you need to use the TK-78K0R/KE3L are selected.)
"Explain" area displays an explanation of the selected product.
To change the installation destination, click [Browse…].

When all the settings are completed, click [Install…].

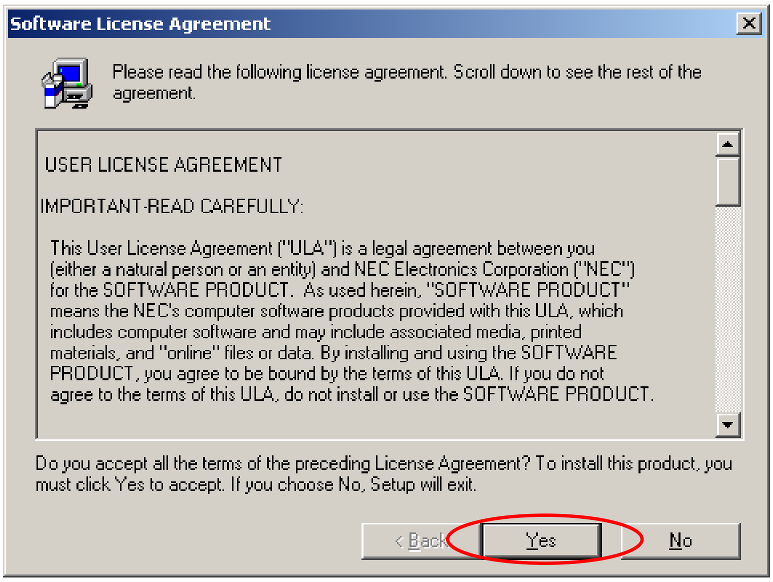
* In this document, it is assumed that users install the programs under "NEC Electronics Tools" directory (default installation directory). Users can find the tools by selecting "Start Menu" -> "Programs" -> "NEC Electronics Tools".



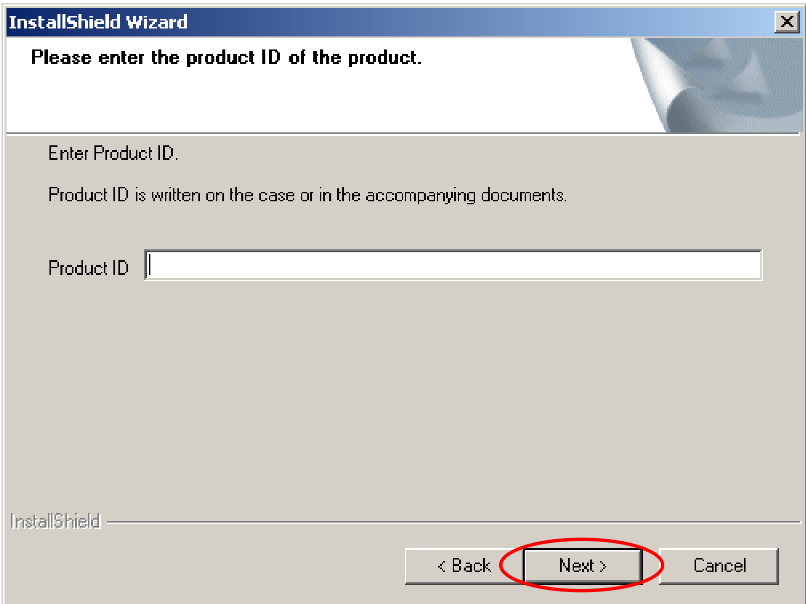
④ Click [OK] when "Install" comfirmation dialog box is opened.



⑤ Read "software license agreement" and click Yes for continuing the installation.
To stop the installation, click No .

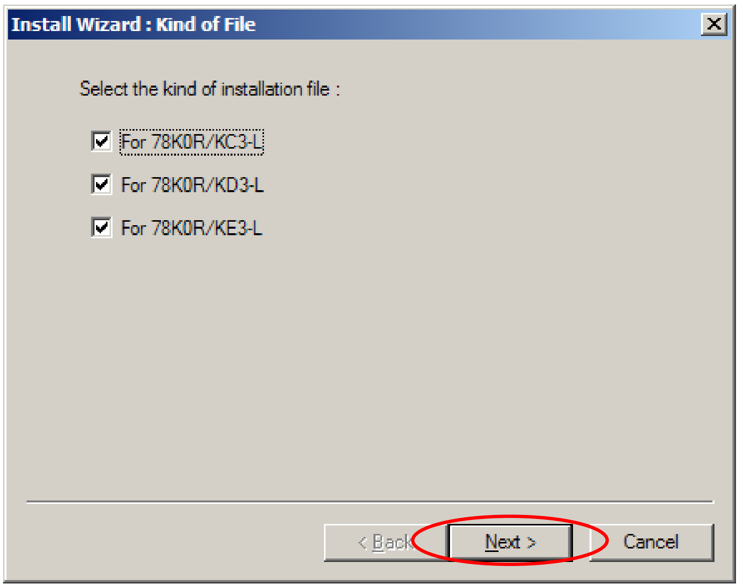


⑥ Enter the product ID, and click Next .
＊ The product ID is available on the other sheet.

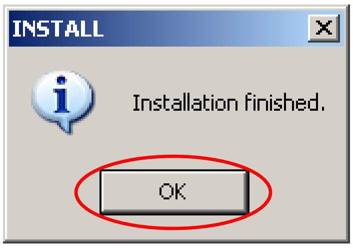


⑦ It starts copying the files.

⑧ Click [Next] when "Select Files" installation wizard dialog opened.



⑨ When the installation is completed, the following dialog opens.
Click [OK] .



⑩ "NEC Electronics Starter Kit Virtual UART" USB driver must be installed on PC before you connect to TK-78K0R/KE3L.

Install the USB driver by referring "1.3 Installation of USB Driver".

Notes on the installation authority

To install this tool in Windows 2000 or XP, the authority of an administrator is necessary. Therefore, please login as an administrator.

Notes on the install-directory

Please do not use 2-byte characters, such as umlaut in the directory name, where the product is to be installed.

Note on the version of Windows

If the language of the Windows is not English, a file transfer error during installation might be observed. In this case, please abort the installation in the language, and re-install it in an English version of Windows.

The identical problem may be observed, if a language other than English is specified as the system language in the “Regional Settings Properties” tab.

Limitation

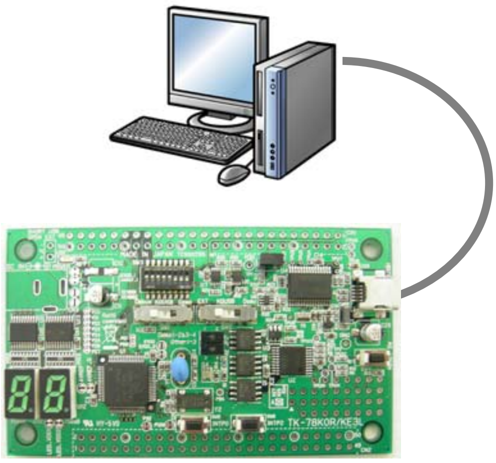
Assembler RA78K0R and C compiler CC78K0R limit the object size to 64 Kbyte.

1.3 Installation of USB Driver

"NEC Electronics Starter Kit Virtual UART" USB driver must be installed on PC before you start using the TK-78K0R/KE3L.
Please, follow the instruction below to install the driver.

"Starter Kit USB Driver" must be installed on the PC. If not, please refer to "1.2 Installation of Development Tools" to install the driver first.

CAUTION:
Do not use a USB hub for connecting TK-78K0R/KE3L.



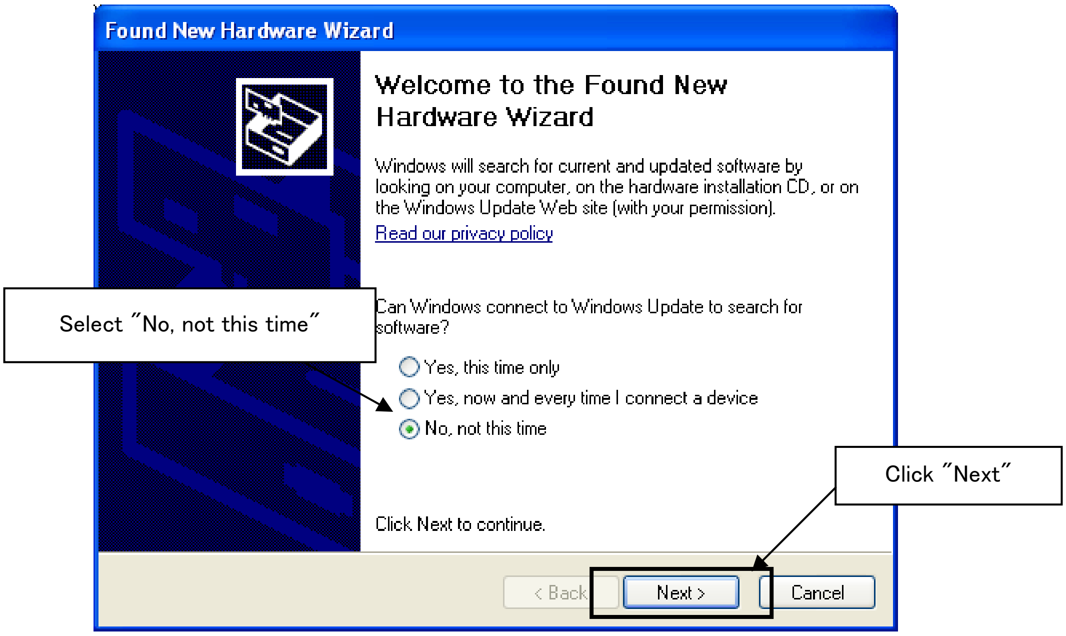
Depending on the version of Windows OS, the installation will be differed. Please check your Windows version, and follow the instructions

- Windows XP -> "1.3.1 Installation on Windows XP"
- Windows 2000 -> "1.3.2 Installation on Windows 2000"

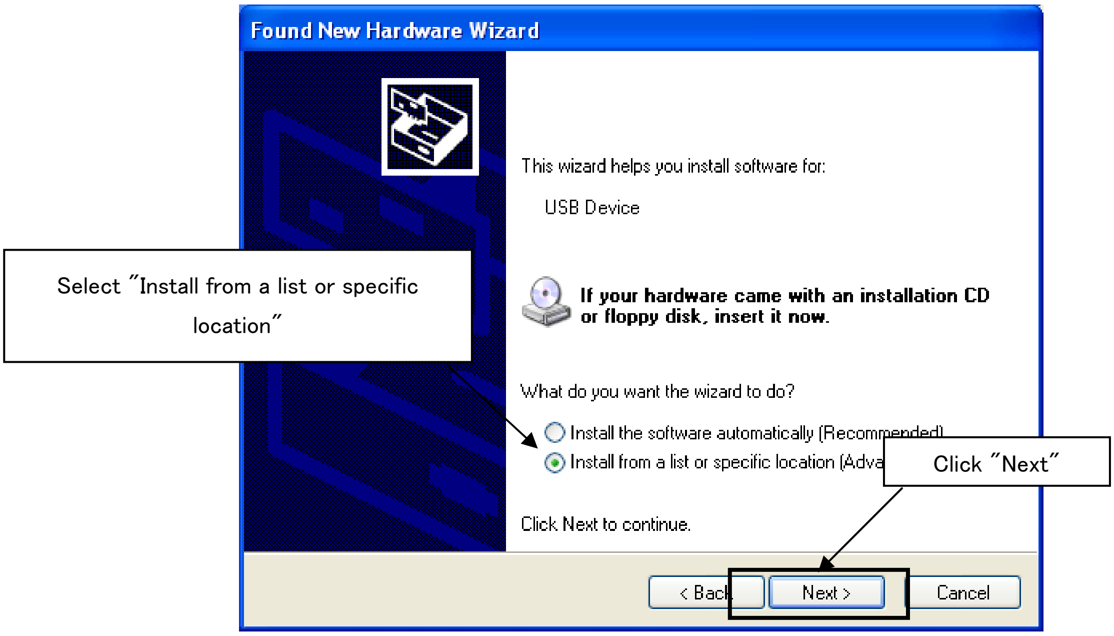
After the installation, go to "1.3.3 Completion of USB Driver Installation"

1.3.1 Installation on Windows XP

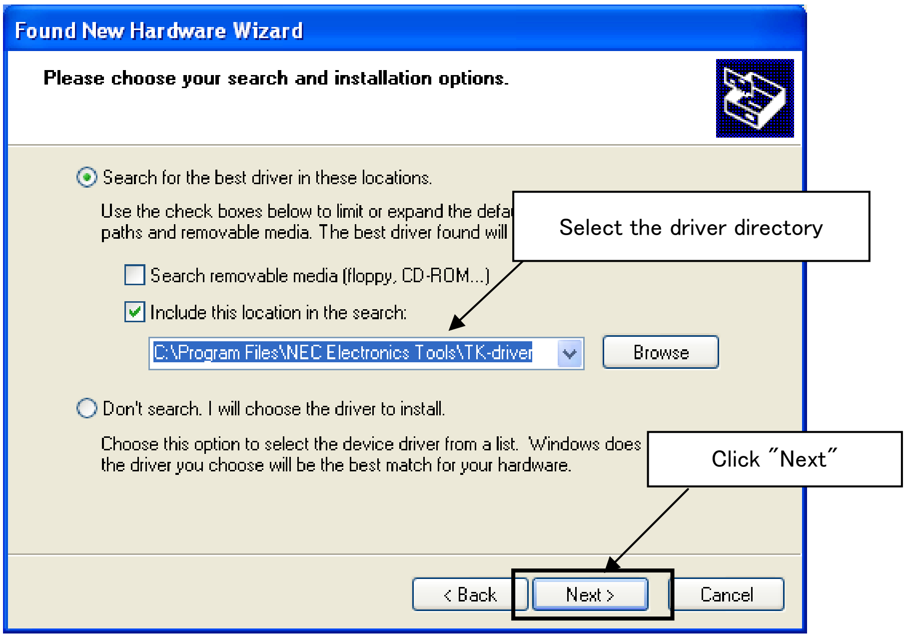
1. Once the TK-78K0R/KE3L is connected with USB, the "Found New Hardware Wizard" will be started.
   Select "No, not this time" and click Next > .



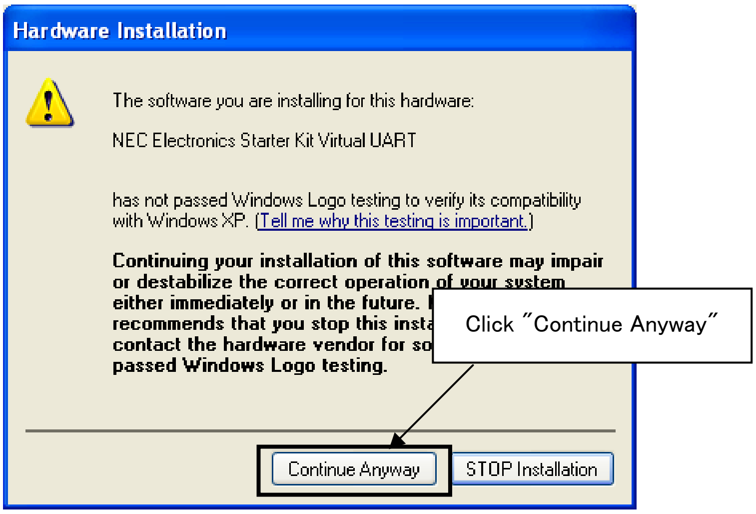
2. Select "Install from a list or specific location" and click Next > .



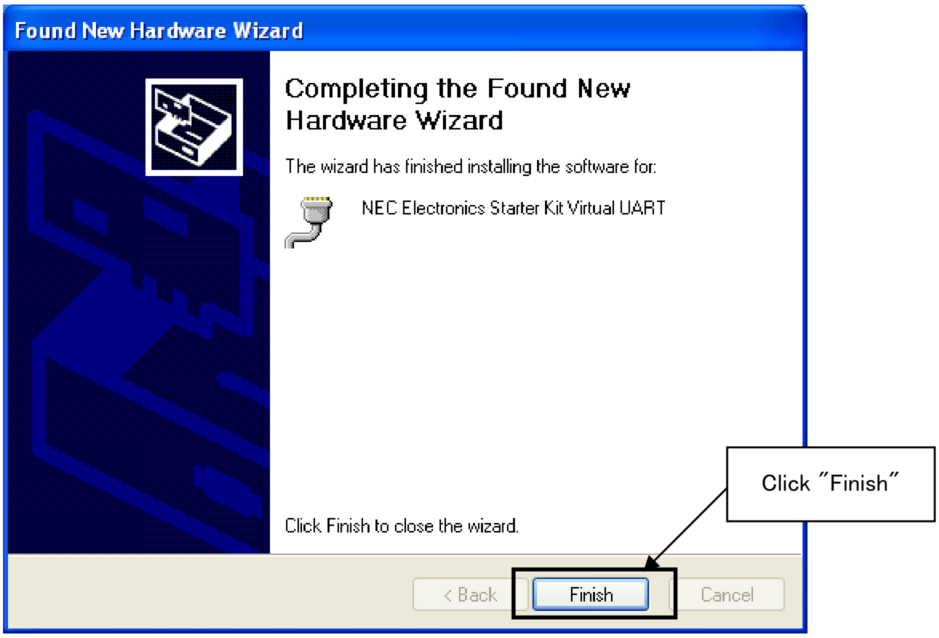
3. Select "Search for the best driver in these locations.", check "Include this location in the search:", and then click "Browse..." to select the driver directory path. The path should be "C:¥Program Files¥NEC Electronics Tools¥TK-driver" as default installation. If the installation directory is not default, then select "TK-driver" under the installation directory.
Click [Next >] .



4. If the following dialog is opened, click [Continue Anyway] .



6. The installation of "NEC Electronics Starter Kit Virtual UART" driver is completed.
   Click [Finish] .



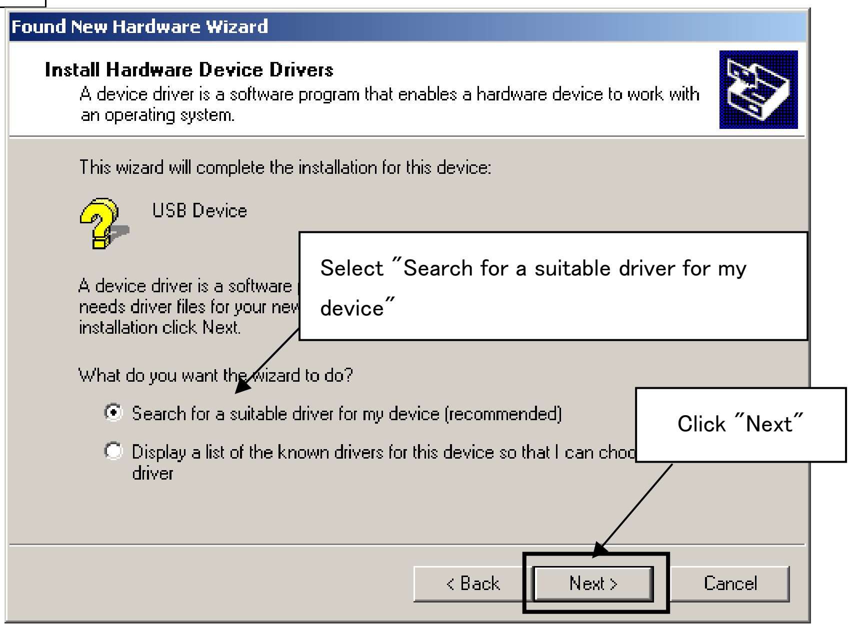
7. Go to "1.3.3 Completion of USB Driver Installation".

1.3.2 Installation on Windows 2000

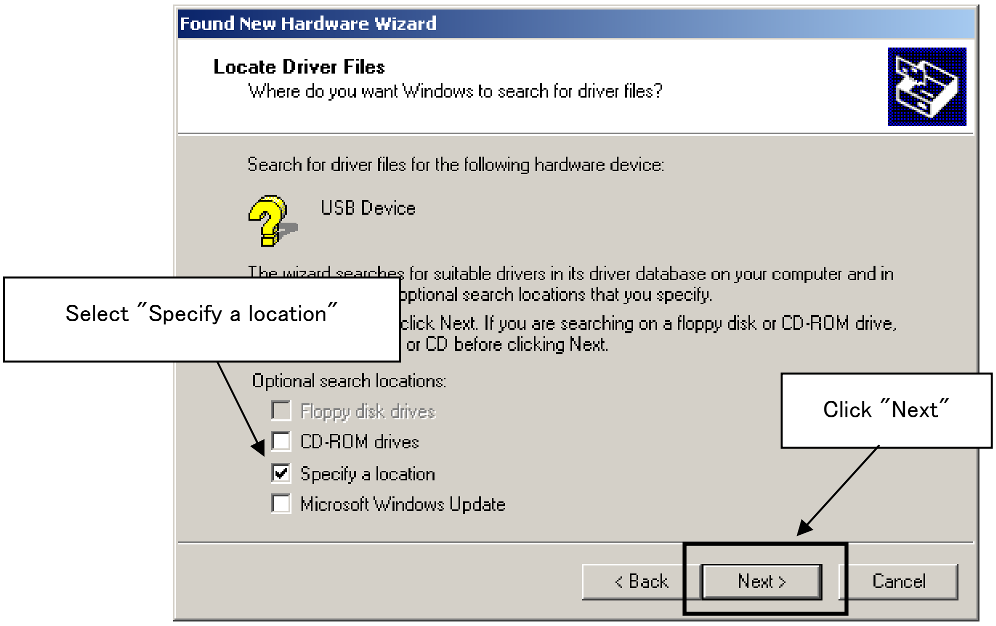
1. Once the TK-78K0R/KE3L is connected with USB, the "Found New Hardware Wizard" will be started.
   Select "No, not this time" and click [Next >].



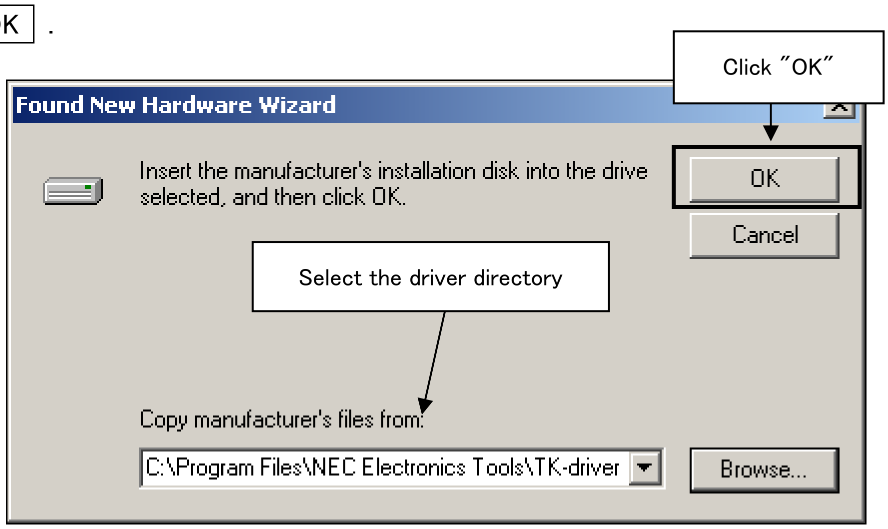
2. Select "Search for a suitable driver for my device".
   Click [Next >].



3. Select "Specify a location".
   Click Next > .



4. Select the driver directory path. The path should be "C:¥Program Files¥NEC Electronics Tools¥TK-driver" as default installation.
   If the installation directory is not default, then select "TK-driver" under the installation directory.
   Click OK .



5. Click [Next >] .



6. The installation of "NEC Electronics Starter Kit Virtual UART" driver is completed.
   Click [Finish] .



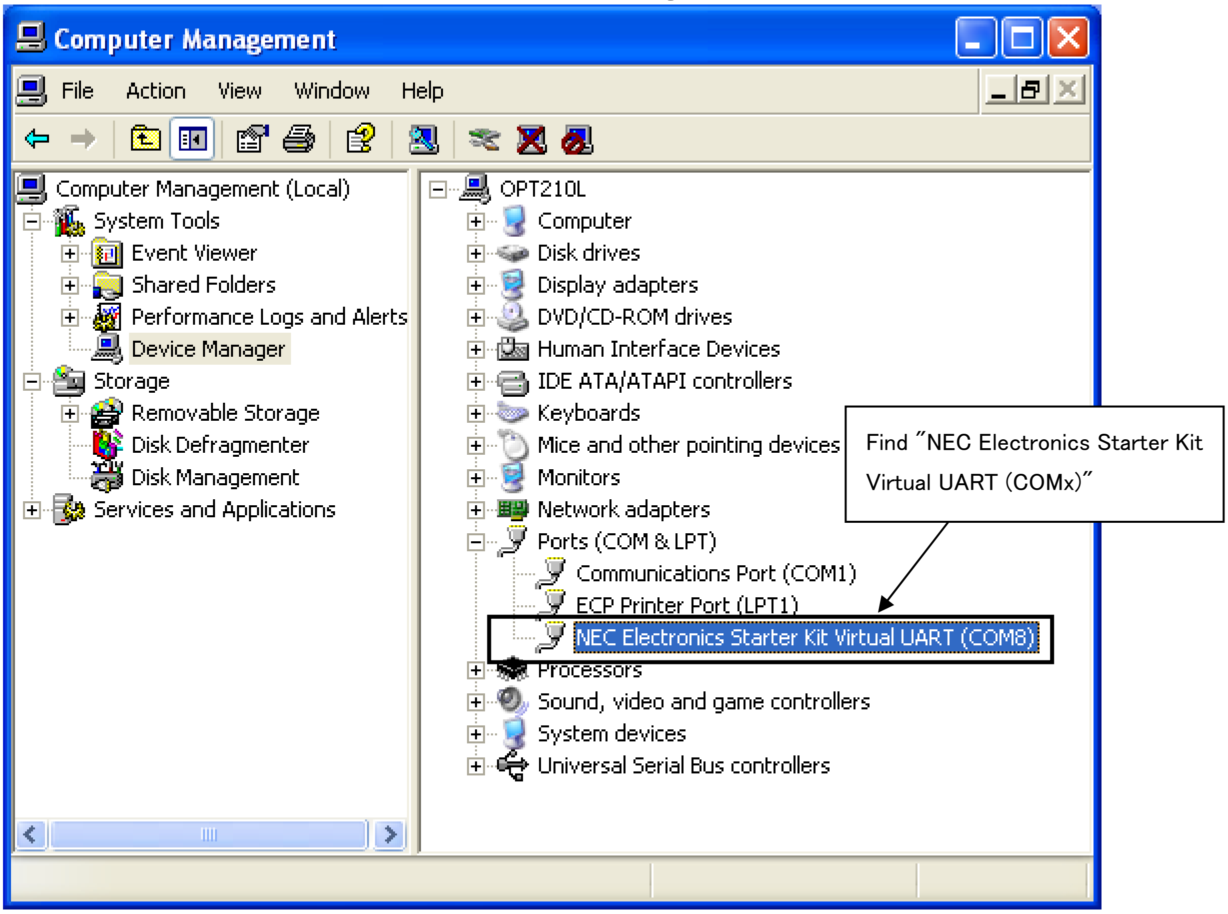
7. Go to "1.3.3 Completion of USB Driver Installation".

1.3.3 Completion of USB Driver Installation

Confirm the USB driver is installed on PC.

Start "Device Manager", and find "NEC Electronics Starter Kit Virtual UART" (without "?" mark) under the "Ports (COM & LPT)".

Device Manager



The screen above shows that the COM port number is "COM8". If ID78K0R-QB-EZ is not in use, you can use this port number for connecting TK-78K0R/KE3L.
When you change the USB port connection, the COM port number will be changed as well.

CAUTION

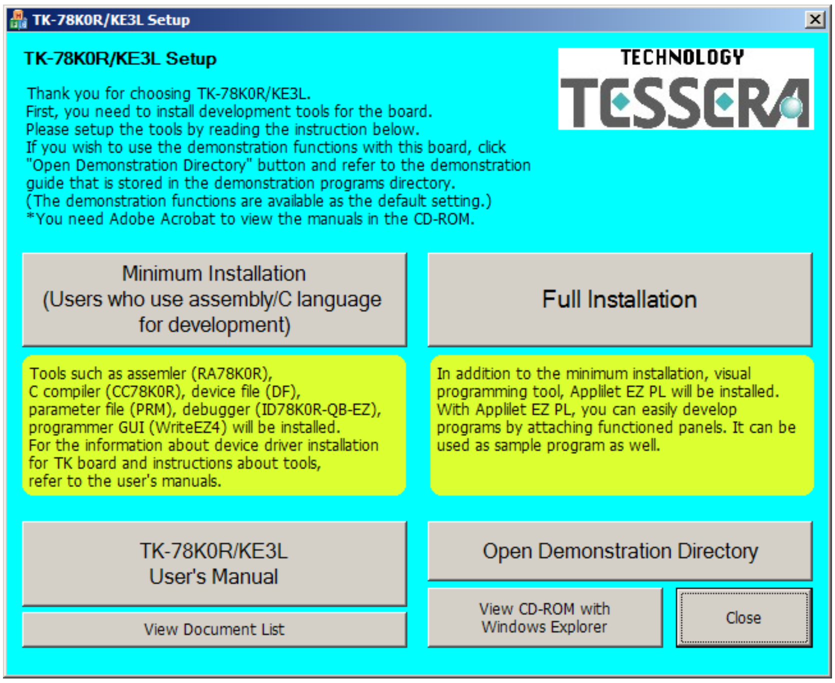
・Do not do "Hardware Modification Scan" when you communicate with the target device.

1.4 Sample Programs

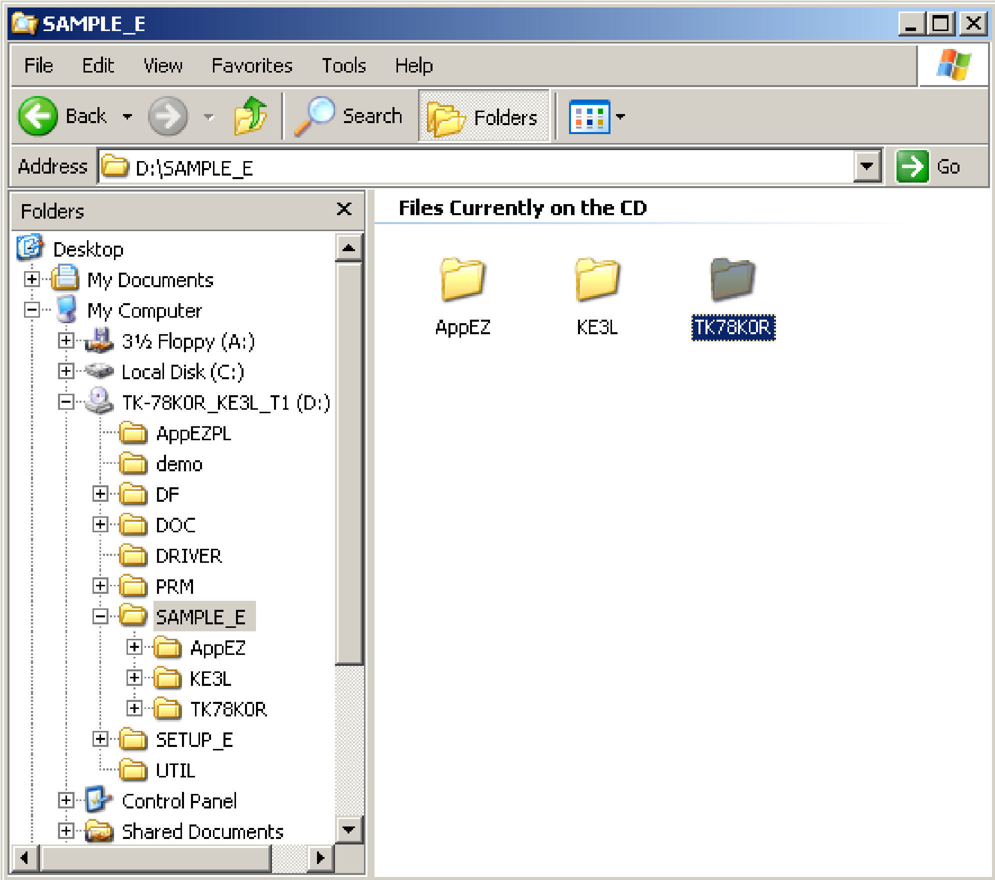
This section explains the overview and preparation of sample programs.

1.4.1 Preparation of Sample Programs

① Please insert the CD-ROM in the drive. The installer will show up automatically.
If it does not start automatically, please initiate it by double clicking the SETUP.EXE.



② Click [View CD-ROM with Windows Explorer].



③ Copy the directory "TK78K0R" under "SAMPLE_E" directory to local PC.

CHAPTER 2 Experiences

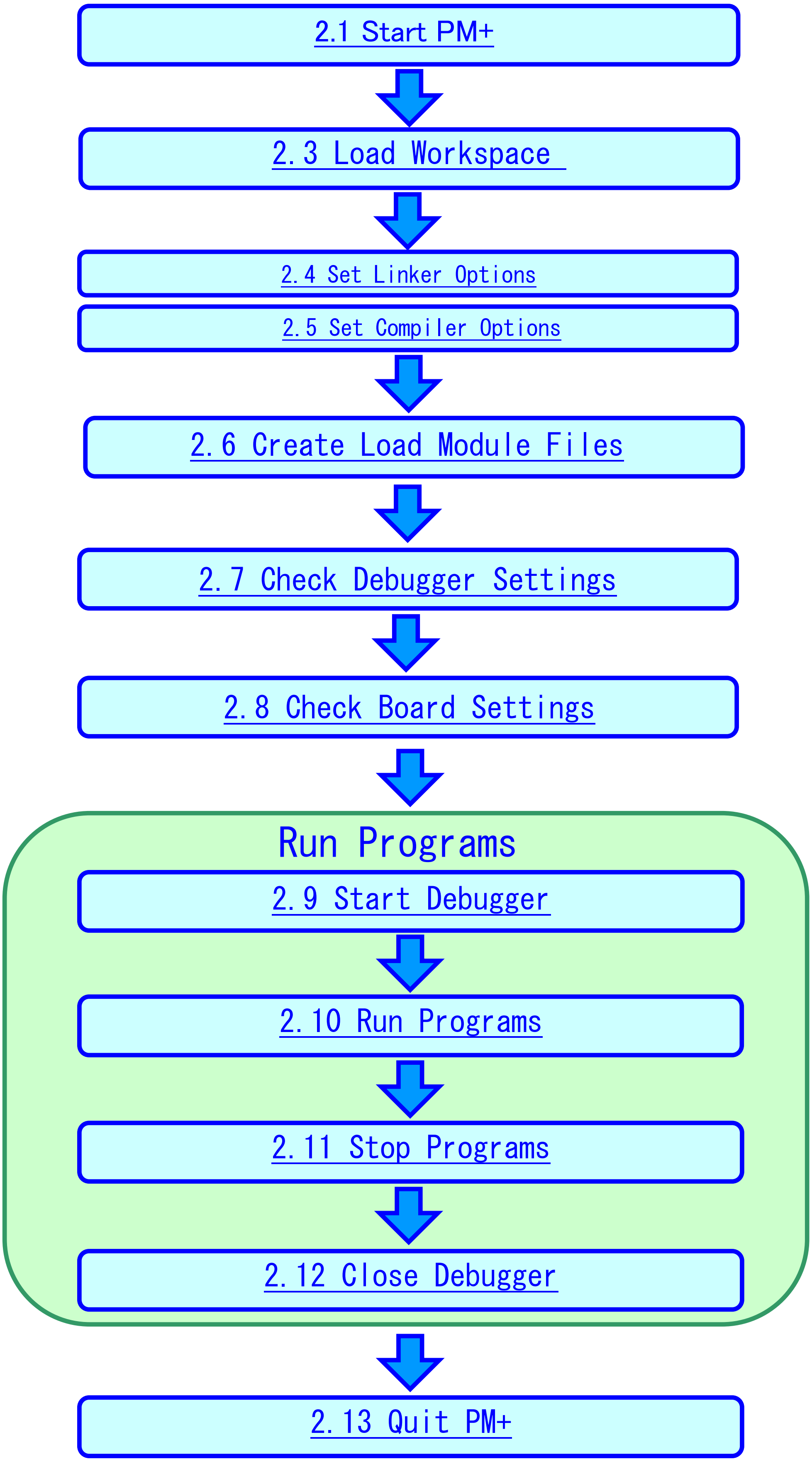
In this chapter, you will experience how to use the development tools with using the sample programs. The development tools are :

- Integrated Development Environment (IDE), PM+
- Integrated Debugger, ID78K0R-QB-EZ

You will use the programs that you prepared in "1.4 Sample Programs", as the sample programs for TK-78K0R/KE3L.

You will be able to understand how to use the development tools and the concept of project files which you need for producing application programs.

The overall steps are as follows:



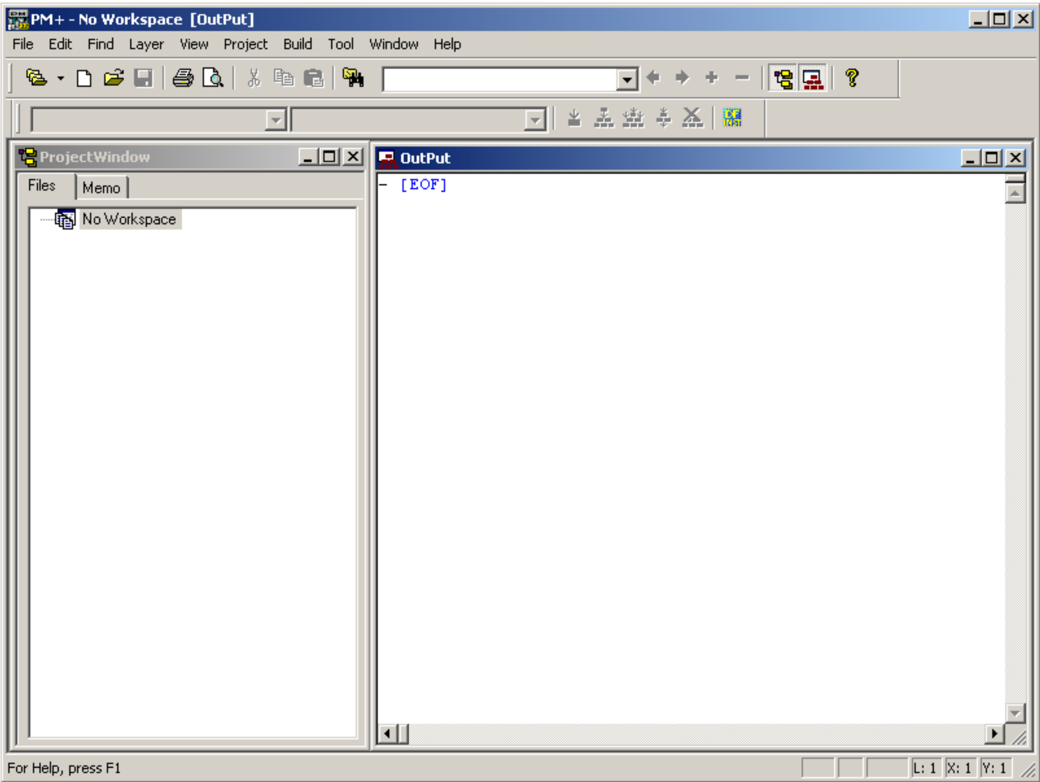
2.1 Start PM+

Let's start using the development tools.

First, start the PM+

Select "Windows Start Menu" -> "Program" -> "NEC Electronics Tools" -> "PM+ V6.31".

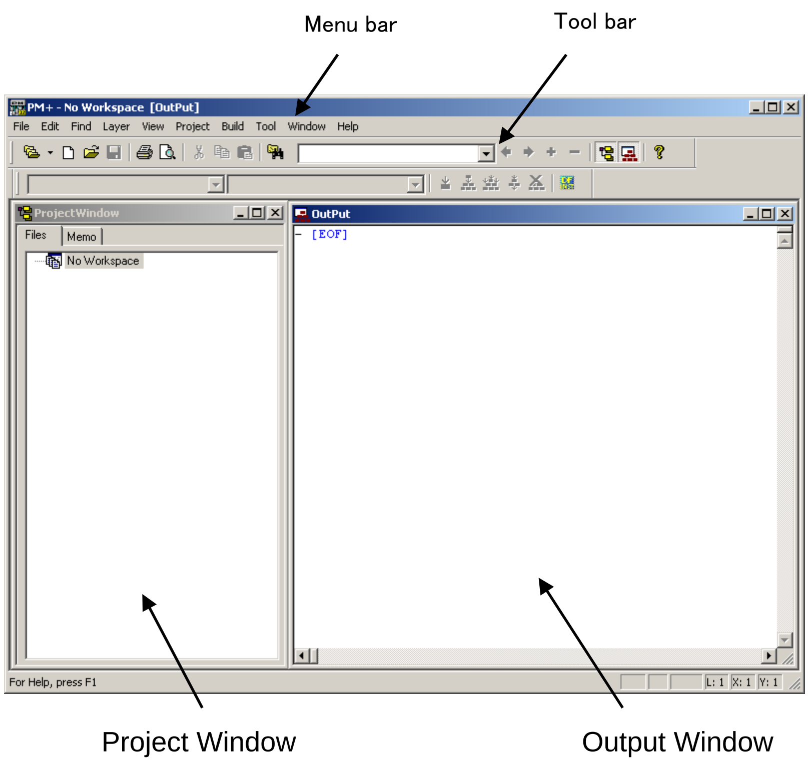
PM+ starts up



2.2 What is PM+

In PM+, application programs and environment setting are handled as a single project, and series of actions such as program creation using the editor, source management, build, and debugging are managed.

Also, one of more project files is managed together as a workspace.



Project window A window in which project names, source files, and include file are displayed using a tree structure.

Output window A window in which the build execution status is displayed.

For details regarding menu bars and tool bars, refer to "Help" menu in PM+.
"Help" on menu bar , then "PM+ Help"

What is a project?

A project is the unit that is managed by PM+. A project refers to an application system and environment development based on PM+.

PM+ saves project information in a "project file".

What is a project file?

A project file contains project information that includes the source files, device name, tool options for compiling, editor, and debugger information.

The file name format is "xxxxx.prj".

Project files are created in the directory you specifies when you create a new workspace.

What is a project group?

A project group is a group comprised of a number of projects in an application system.

The target device of each project must be the same within a project group.

What is a workspace?

A workspace is the unit used to manage all the projects and project group required for one application system.

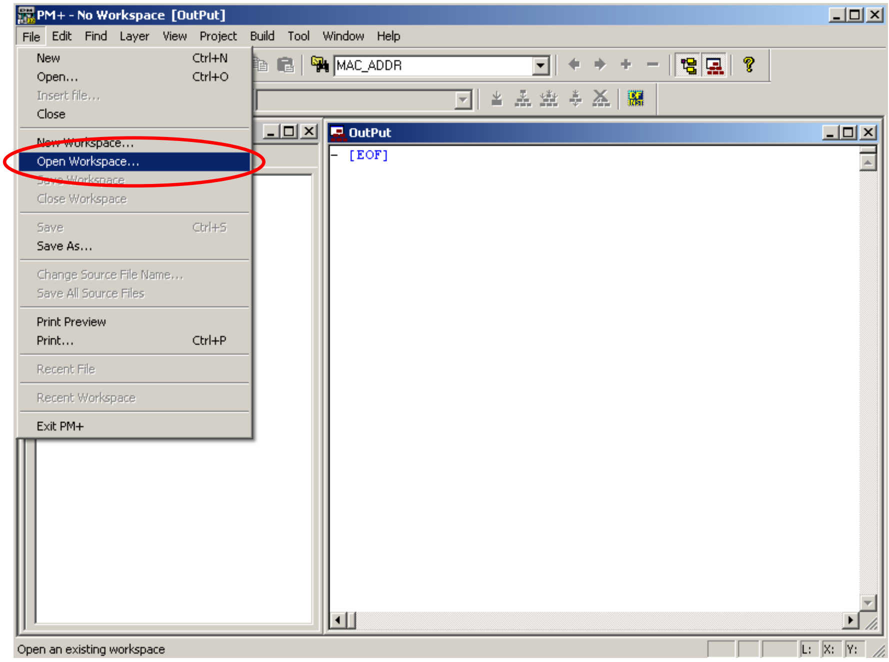
A workspace file contains one or more project files.

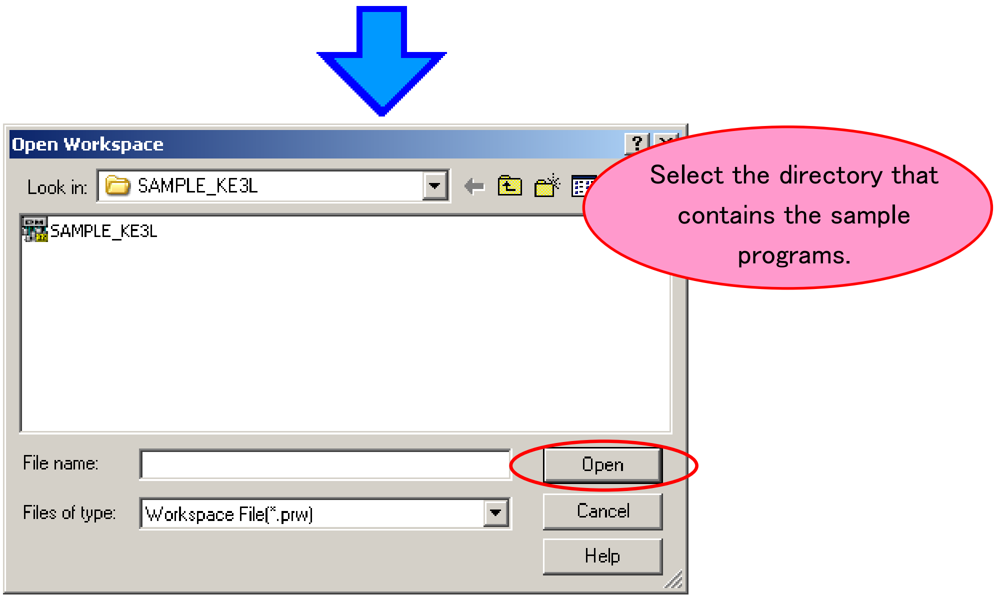
The file name format is "xxxxx.prw".

2.3 Load Workspace (project)

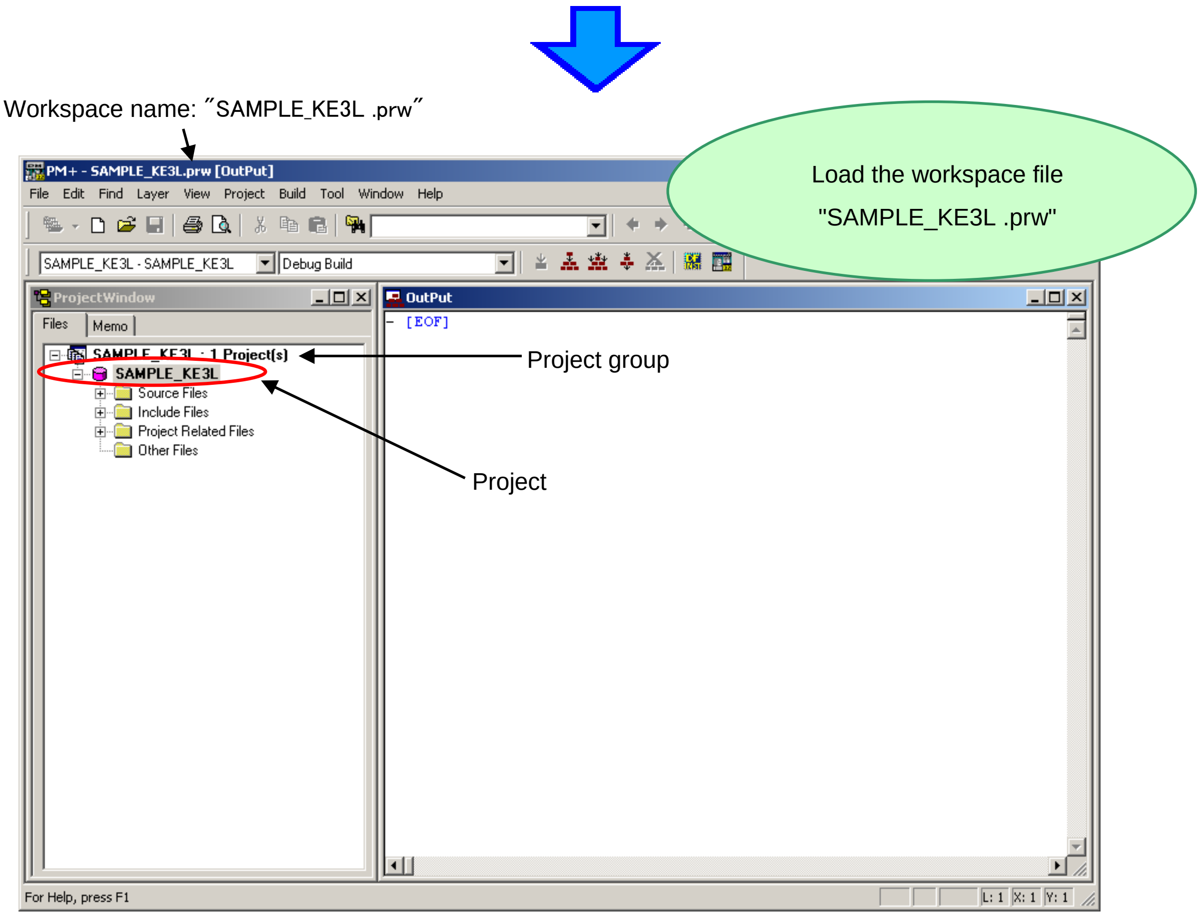
In this section, you will use the workspace that you created in "1.4 Sample Programs"
For creating a new workspace, refer to "Chapter 5 Other Information".
The workspace has information about the build environment for the sample programs.

Select "File" on menu bar and "Open Workspace…".
Then, select "SAMPLE_KE3L .prw" under the directory "TK78K0R¥SAMPLE_KE3L¥".





Select "SAMPLE_KE3L .prw", then click Open .



The workspace file "SAMPLE_KE3L .prw" contains one project called "SAMPLE_KE3L". You will use this project "SAMPLE_KE3L".

CAUTION:

Please ignore when you get a prompt saying "files could not be found". This may occurred when the installation directory is not a default.

2.4 Set Linker Options

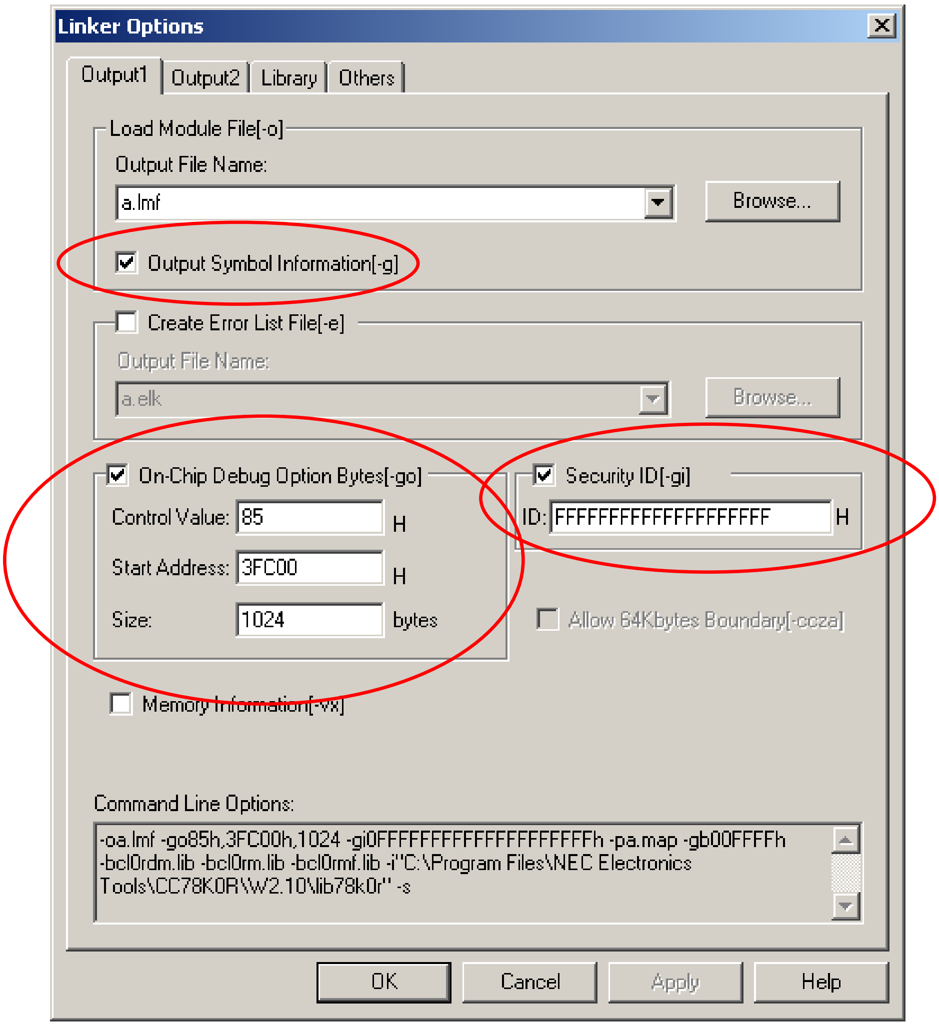
The linker options have been set by the project file. However, some option settings will be covered in this section because the linker option settings are important for debugging. Following three settings are covered specifically.

- Outputs from debugging
- On-chip debug (Disable/Enable, security ID)
- Watchdog timer

Select "Tools" on menu bar, then "Linker options....".

2.4.1 "Output1" Tab

Select "Output1" tab on "Linker Options" window, and see following settings.



- Load Module File settings

Check "Output Symbol Information".

This enables to do source level debugging (setting break points, monitoring variables in watch window, etc).

Also, you can specify the load module file name.

- On-Chip Debug Option Byte

Check "On-Chip Debug Option Byte". Enter "85" in "Control value". This setting enables the on-chip debugging function of the microcontroller.

* For details of "Control value", refer to the user's manual of 78K0R/Kx3-L (U19291E).

See "Start address" is set to "3FC00", and "Size" is set to "1024".

These settings reserve the memory address area for the monitor program (the flash memory area that the debugger uses for on-chip debugging).

In this case, the "Control value" is allocated to the address of C3H in flash memory, and FFH is set to the next address. Because of this, the following areas could not be set the segments.

<Address area that reserved by on-chip debugging>

- 2H, 3H
- CEH-D7H
- From the address set in "Start address" to the byte set in "Size"

- Security ID

Check "Security ID", and enter the security ID which is a unique ID code (10 bytes) to authenticate when the debugger is launched.

The security ID is stored in the flash memory (C4H-CDH), and checked if it is the same as the code entered in Linker options dialog when the debugger is launched.

The debugger will not be launched when the security ID is unmatched. By using this function, you can secure the programs from leaks.

If you do not need to set the security, it is recommended to set the security ID "FFFFFFFFFFFFFFFFFFFF" as this is the initial code.

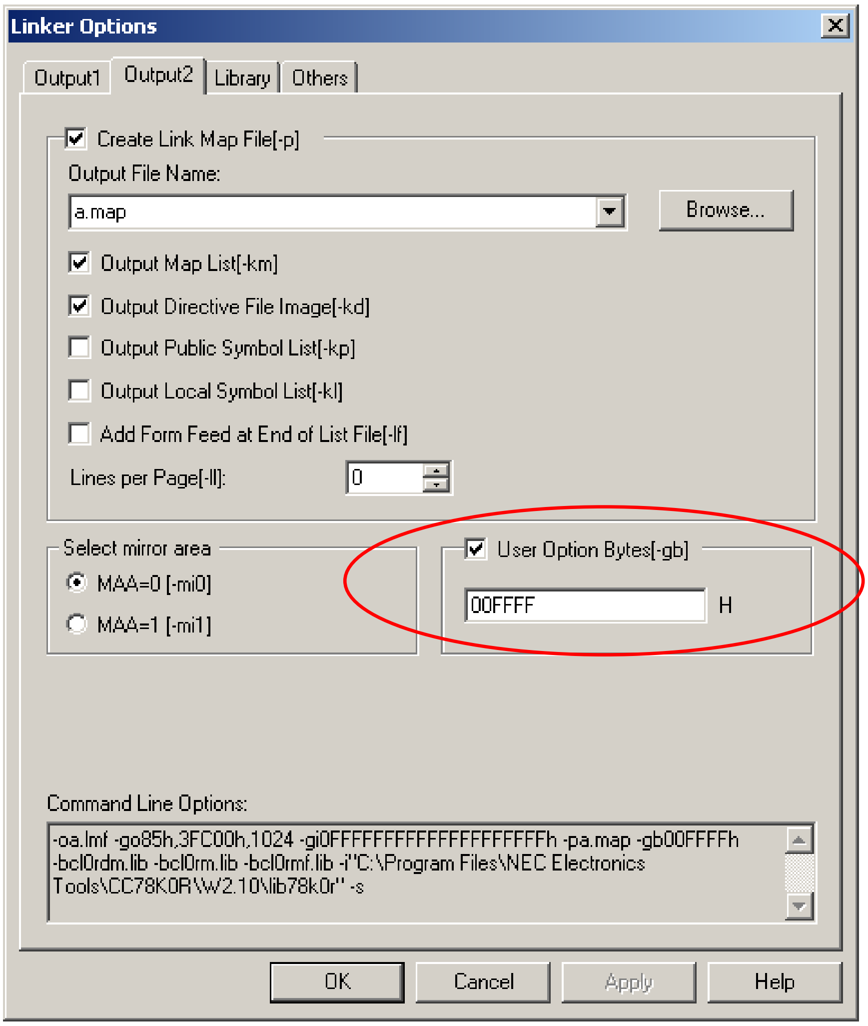
If you forget about the security ID (stored in the address of C4H-CDH) or if you set wrong on-chip debug option byte, you will not be able to use the debugger (ID78K0R-QB-EZ).

In this case, you can erase 78K0R/KE3-L built-in flash memory with "WriteEZ4" to connect with ID78K0R-QB-EZ.

For details, refer to "5.4 Erase microcontroller built-in flash memory".

2.4.2 "Output2" Tab

Select "Output2" tab on "Linker Options" window, and see following settings.



- User Option Byte

Check "User Option Byte", and then enter "00FFFF". Here, you can do the setting of watchdog timer, low-voltage detector, and system reserved memory area. The 3 bytes you entered are stored at C0H-C2H on flash memory:

- C0H: setting for watchdog timer
- C1H: setting for low-voltage detector
- C2H: setting for system reserved memory area (must be set as FFH)

This time, you disabled the watchdog timer and the default start function of low-voltage detector. For details, refer to the user's manual, 78K0R/Kx3-L (U19291E).

2.5 Set Compiler Options

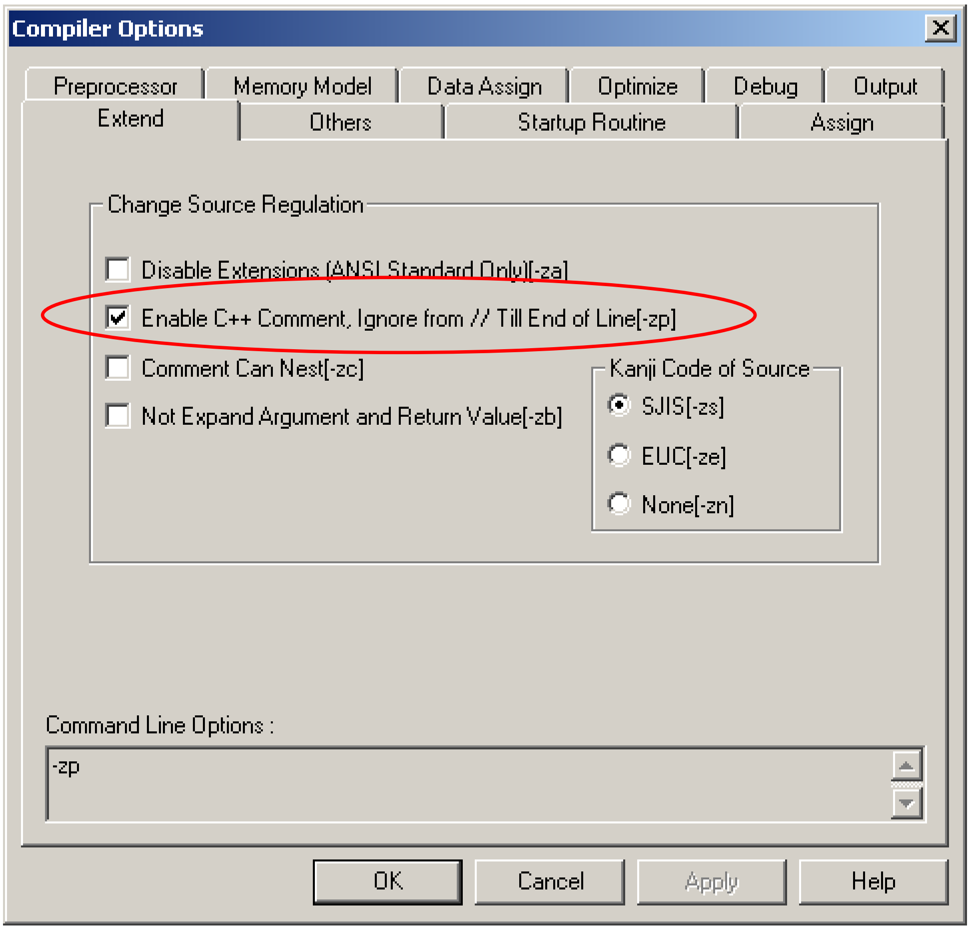
The compiler options have been set by project file. However, because some compiler options are useful, following two settings are covered specifically in this section.

- Enable C++ comments
- Use multiplier and divider

Select "Tools" on menu bar, then "Compiler options".

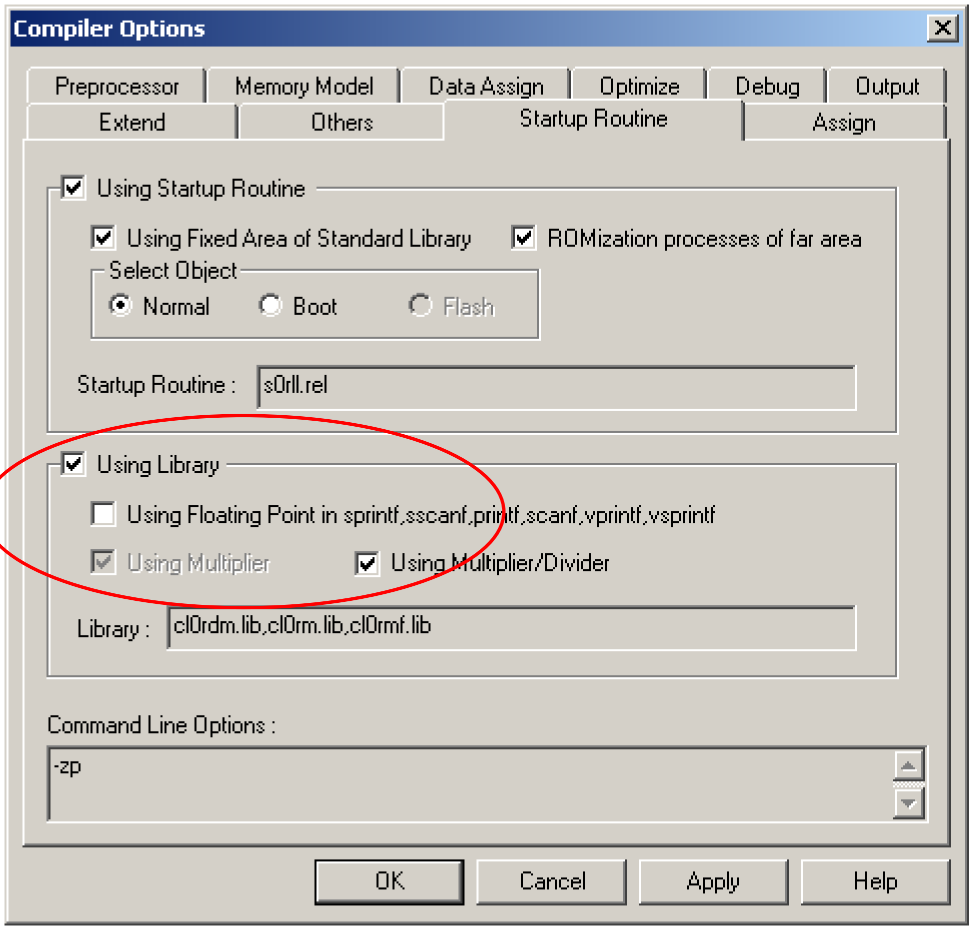
2.5.1 "Extend" Tab

Select "Extend" tab, and check "Enable C++ Comment".
This setting allow you to use the C++ comment using "//".
It is useful feature when developing code.



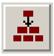
2.5.2 "Startup Routine" Tab

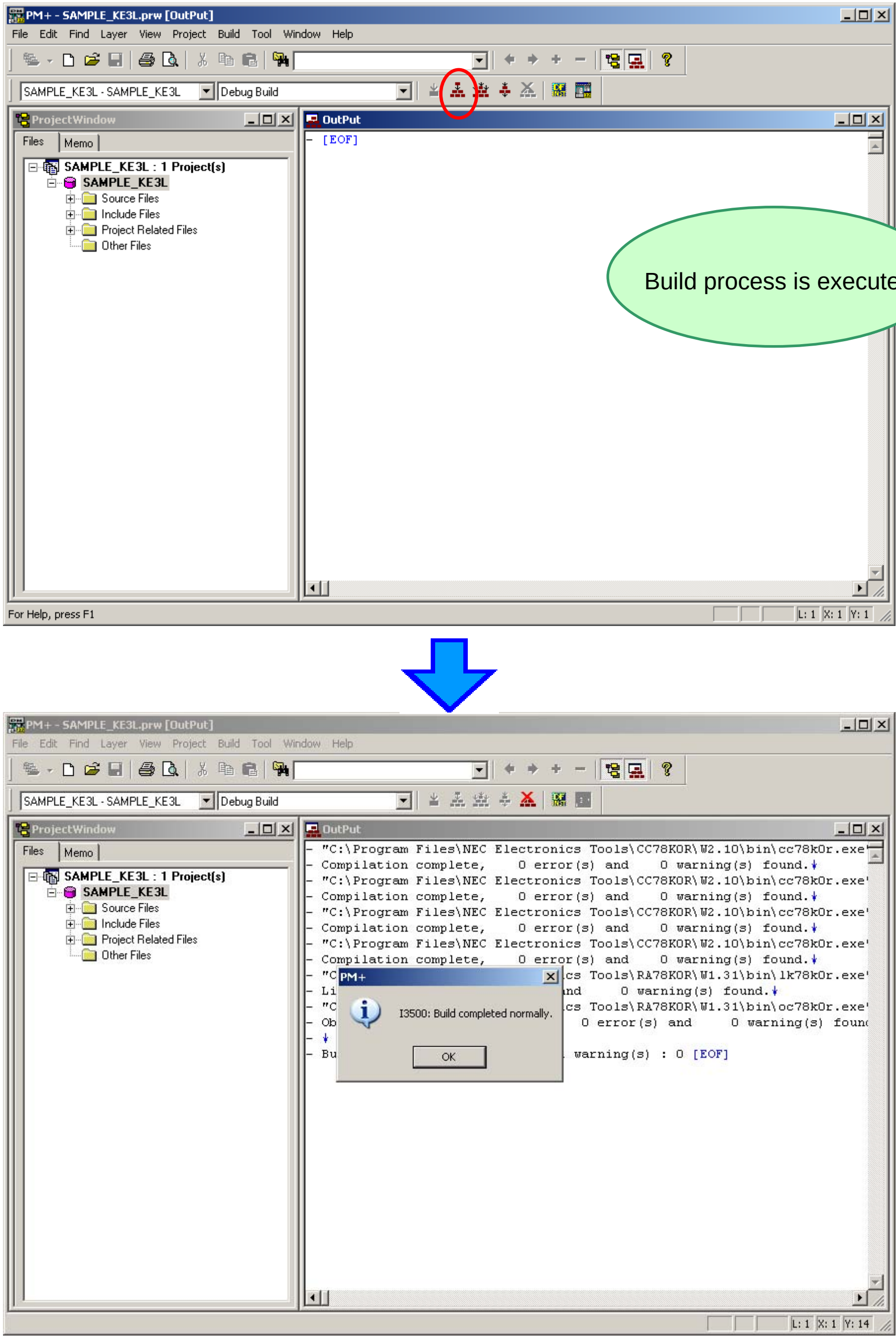
Select "Startup Routine" tab, and check "Using Library" and "Using Multiplier". The 78K0R/Kx3-L has feature of multiplier to increase those calculation speed.



2.6 Create Load Module Files

After developing the source code, you have to create load module files by compiling, assembling, and linking. This process is called build.

Click the build button  , or select "Build" on menu bar, then "Build".



Build has been completed successfully.

What is build?

Build is a function that creates an executable file from source files in a project.

PM+ automatically performs compiling, assembling, linking, and other processing actions.

To reduce the time for the build, PM+ detects and compiles/assembles only the files that have been updated from the previous build process.

What is rebuild?

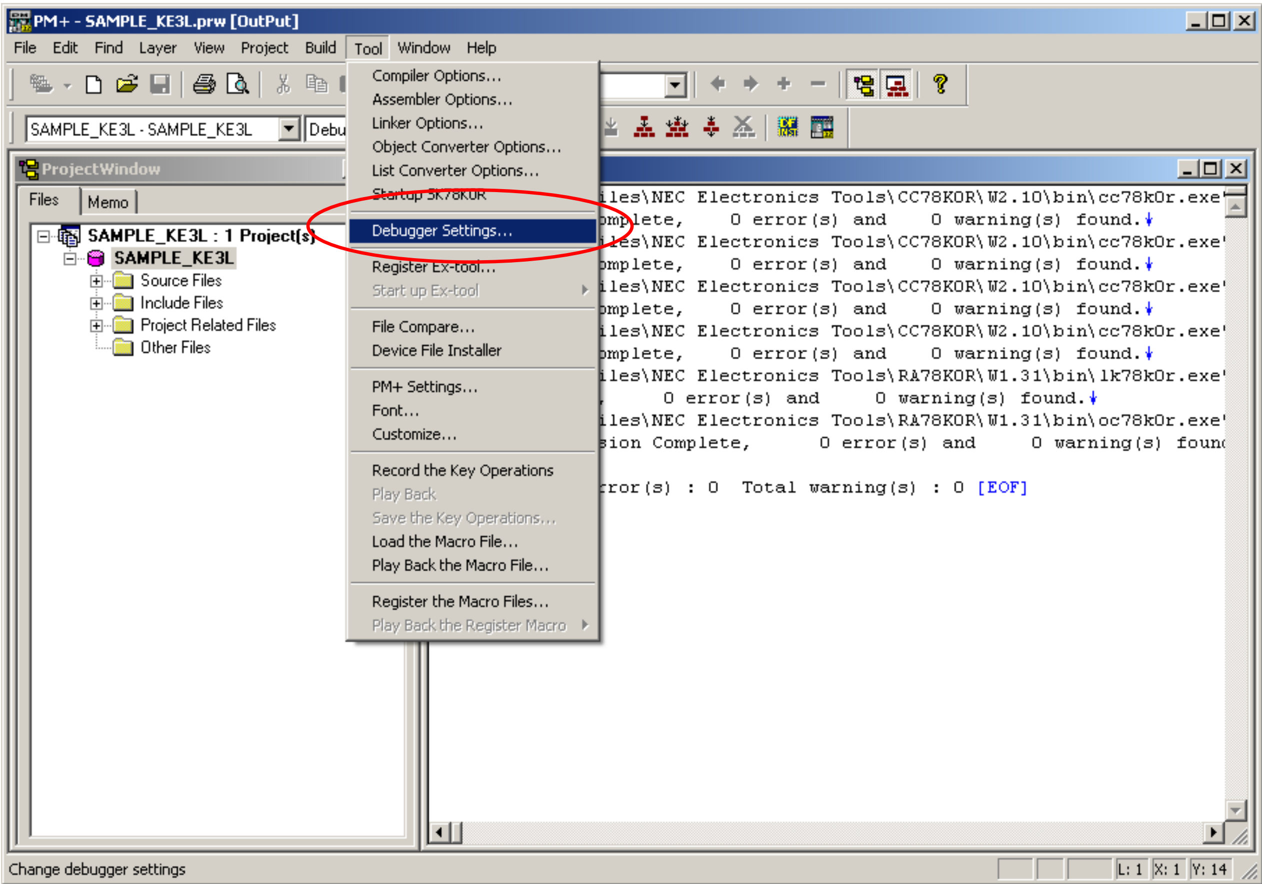
Build compiles and assembles only the source files that have been updated from the previous time, whereas rebuild compiles and assembles all the source files.

When setting, such as compiler options, have been changed, you must rebuild instead of build.

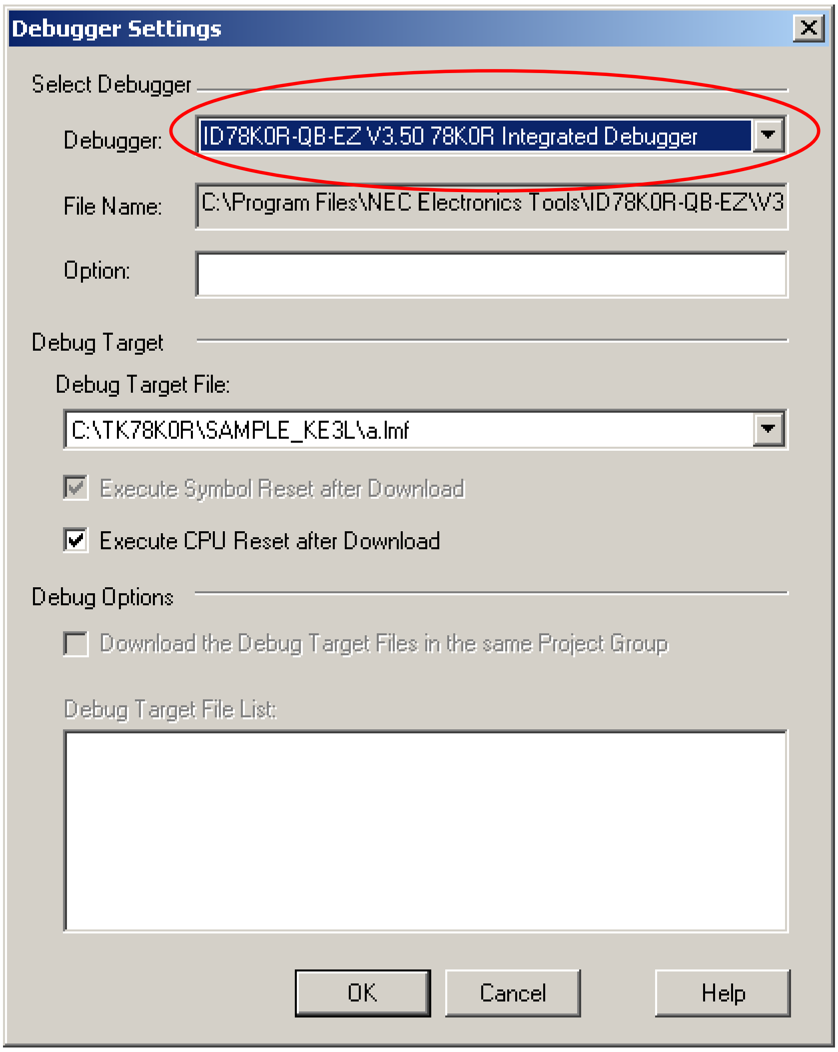
2.7 Check Debugger Settings

After the build, you should configure the debugger settings.
The debugger settings have been set by the project file as well. However, because those settings are important for debugging, some settings are covered in this section.

Select "Tools" on menu bar, then "Debugger Setting…".



Check if "ID78K0R-QB-EZ V3.50 78K0R Integrated Debugger" is selected on "Debugger".

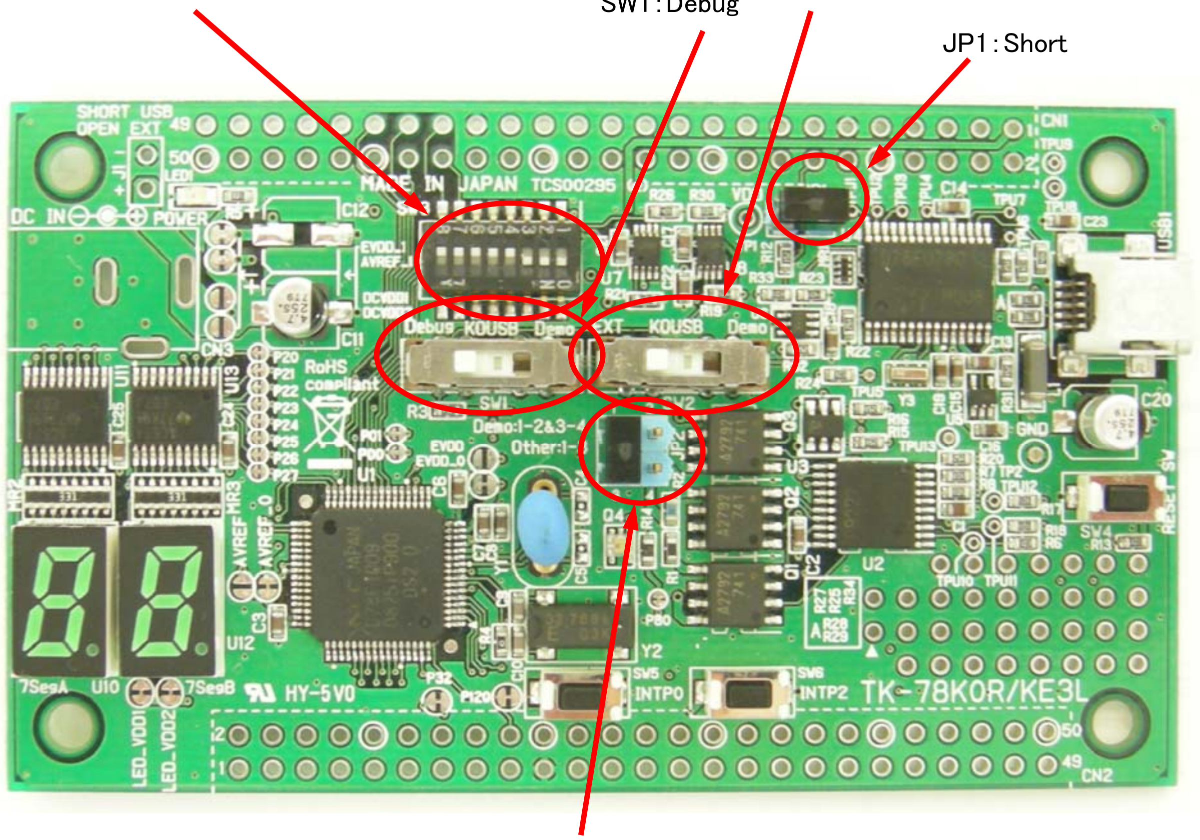


If you cannot select "ID78K0R-QB-EZ V3.50 78K0R Integrated Debugger", select "Project" on menu bar, "Project settings" -> "Tool version settings" -> "Detailsetting" -> then select "ID78K0R-QB-EZ".

2.8 Check Board Settings

Before connecting the PC and the TK-78K0R/KE3L with USB, you should check the setting of SW1,SW2,SW3,JP1 and JP2 on the board.
Set the SW1,SW2,SW3,JP1 and JP2 of the TK-78K0R/KE3L as follows.

SW3 settings							
8	7	6	5	4	3	2	1
OFF	OFF	OFF	OFF	OFF	ON	ON	ON



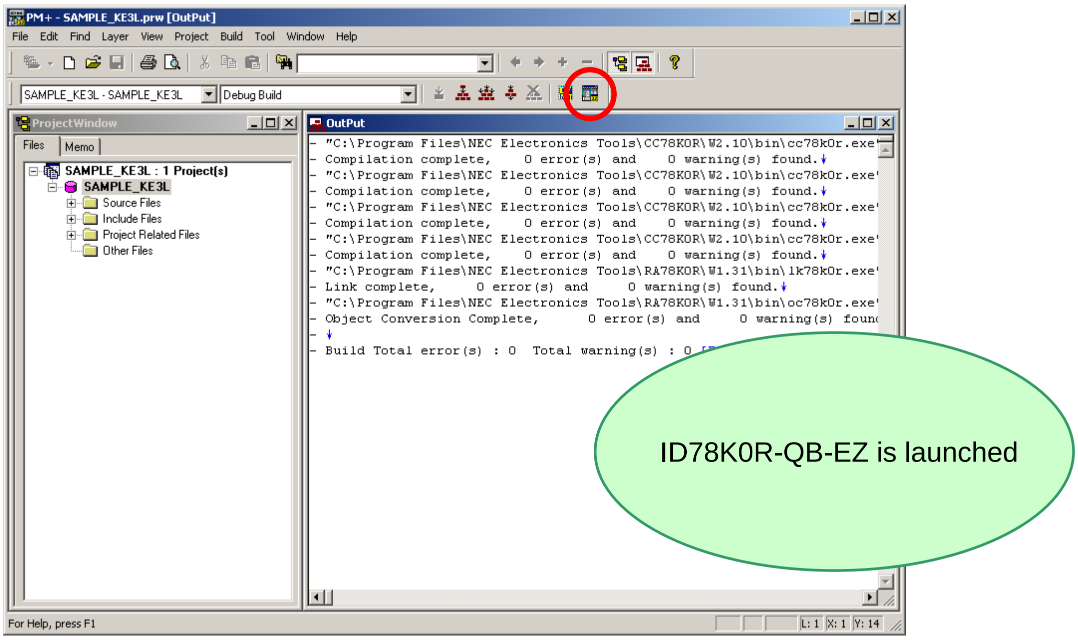
After the switch settings are completed, connect the PC to USB1 on TK-78K0R/KE3L with USB cable. If the "Found New Hardware Wizard" is started, install USB driver with referring "1.3 Installation of USB Driver".

2.9 Start Debugger (ID78K0R-QB-EZ)

Click the debug button  , or select "Build" on menu bar, then "Debug".

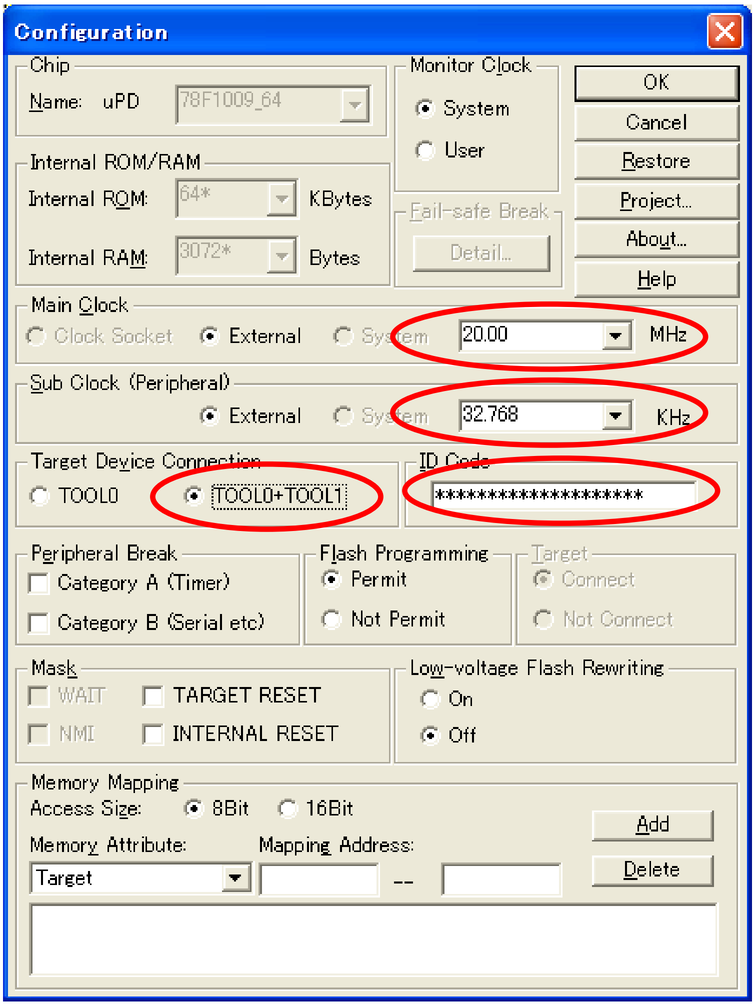
If you do not see the debug button, go to "2.7 Check Debugger Settings" for changing the settings.

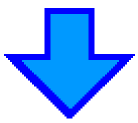
The steps to start the debugger will be explained below.



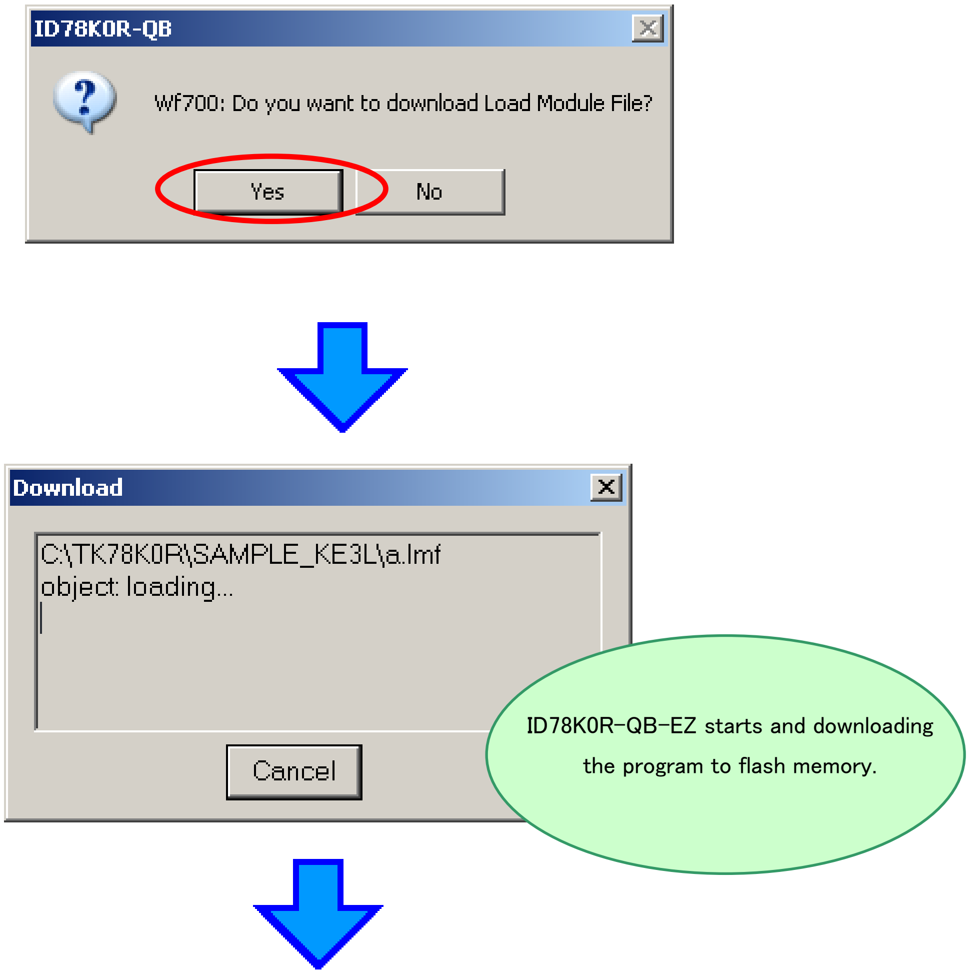
Configuration dialog opens. Set the settings shown below, and then click "OK".

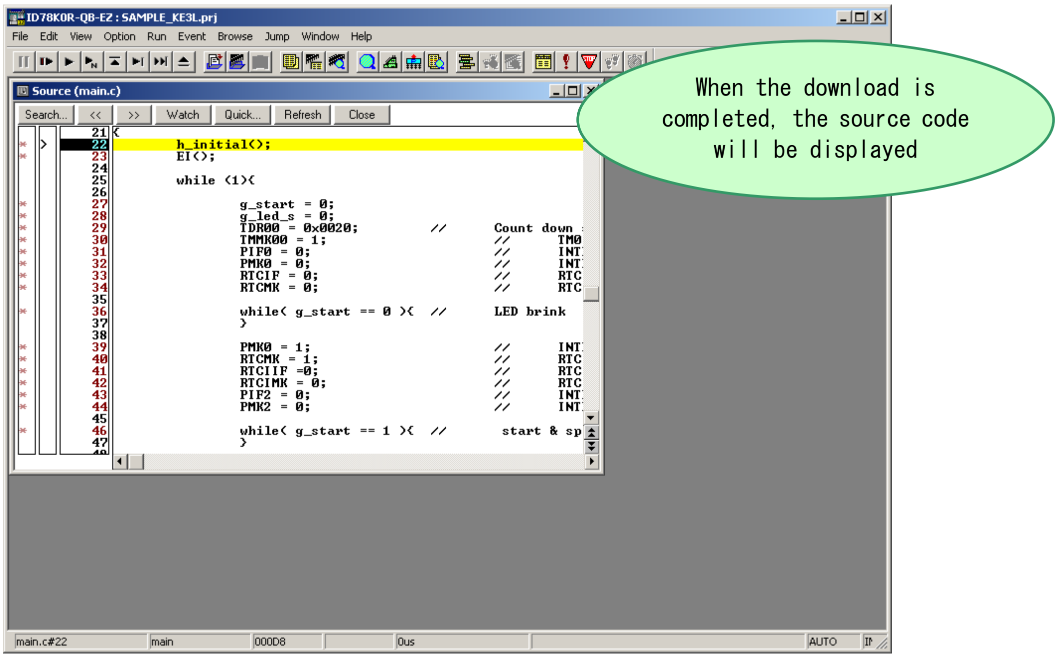
Main Clock	Select "20.00"
Sub Clock	Select "32.768"
Target Device Connection	Select "TOOL0+TOOL1"
ID Code	Enter "FFFFFFFFFFFFFFFFFFFF" (F x20)





Click [Yes] when the confirmation dialog for downloading load module file is opened.





NOTE:

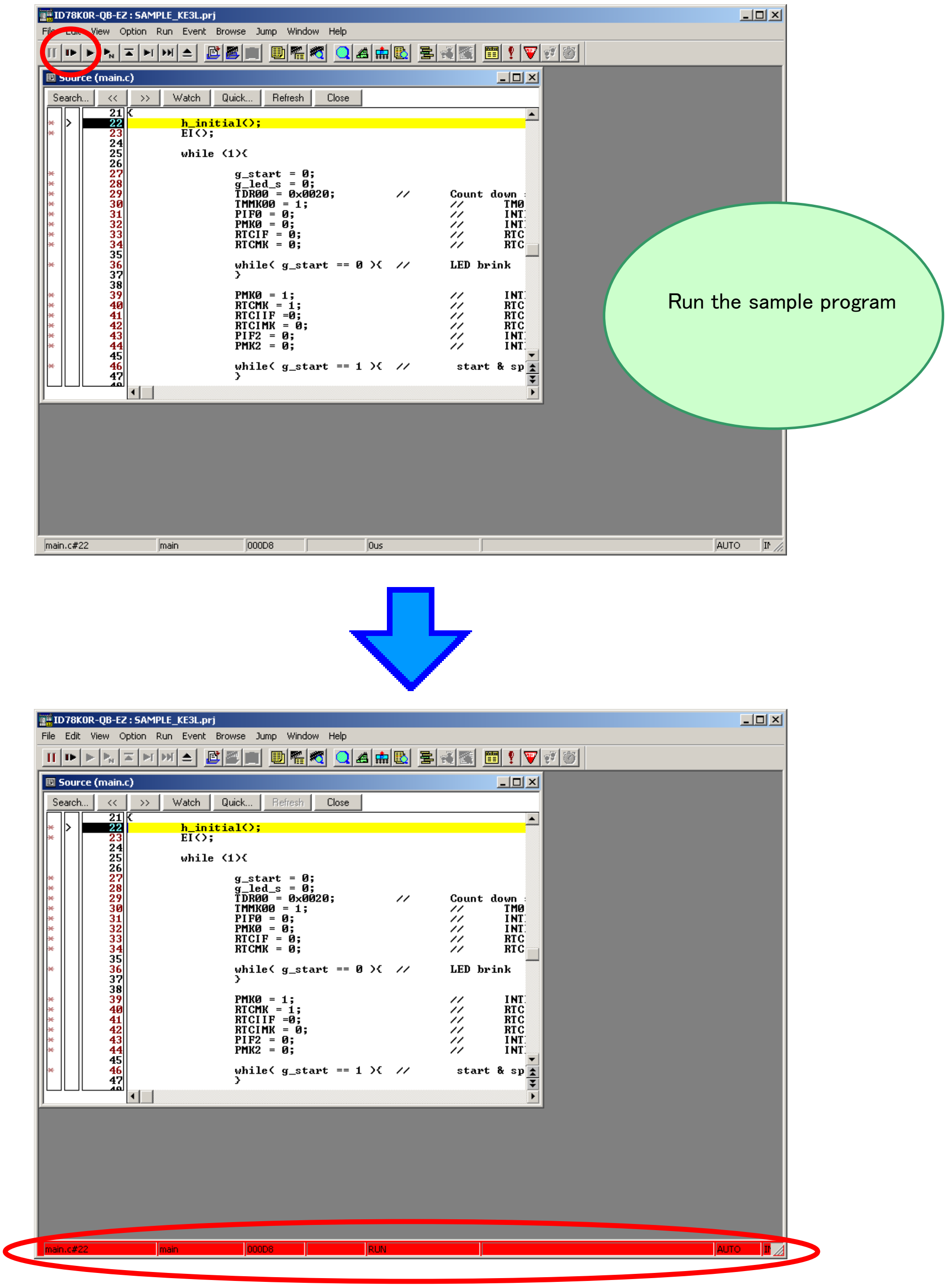
Completion of the download does not mean running the programs. Therefore, even though you press SW1 on the board, it does not make anything happened. To run the demonstration, see "2.10 Run Programs".

2.10 Run Programs

Now, you are ready to run the program.

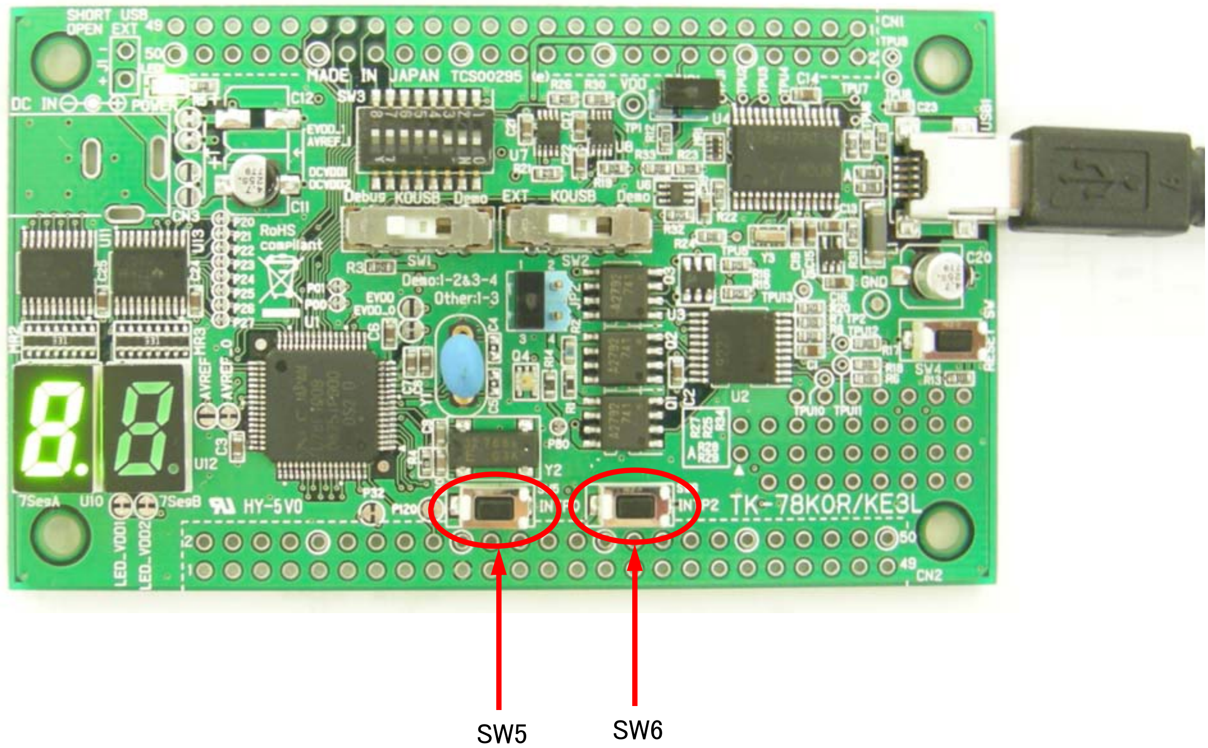
Click the restart button, or select "Run" on menu bar, then "Restart".

The sample program runs.



When programs are running, the status bar will be red.

Check that the LED on TK-78K0R/KE3L is lighted.



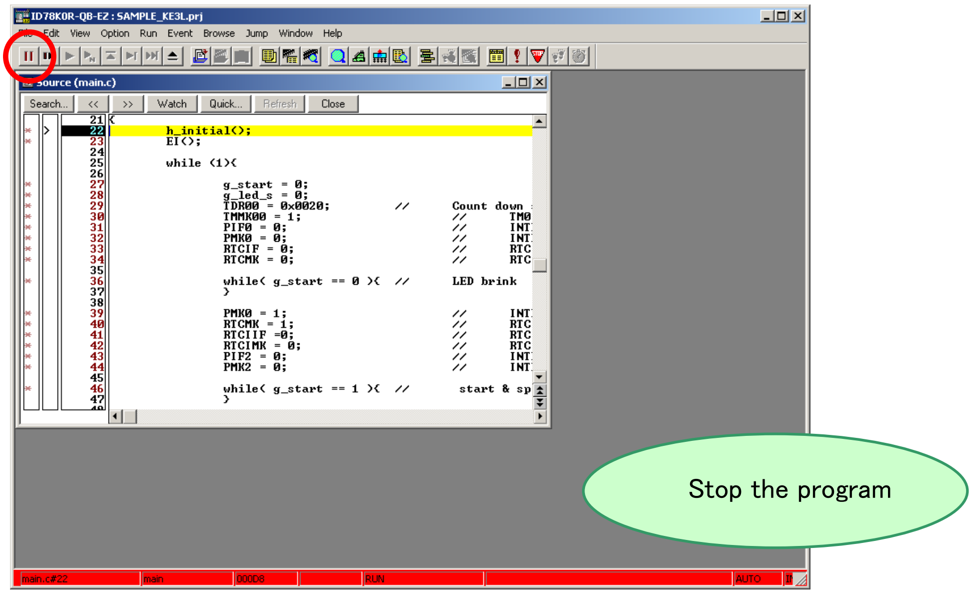
When you press SW5, the segment of 7segLED blinks.
When you press SW6, number is displayed in 7segLED and it starts to count down. After a while, it stops and blinks.

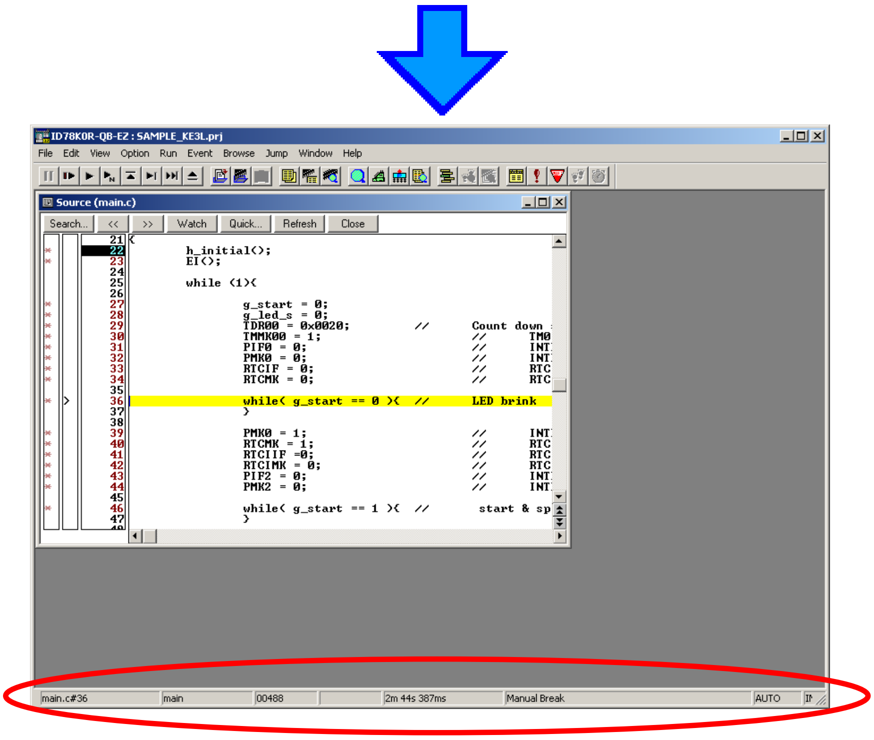
You could confirm the sample program is working.

2.11 Stop Programs

Now, you are going to stop the program.

Click the stop button , or select "Run" on menu bar, then "Stop".

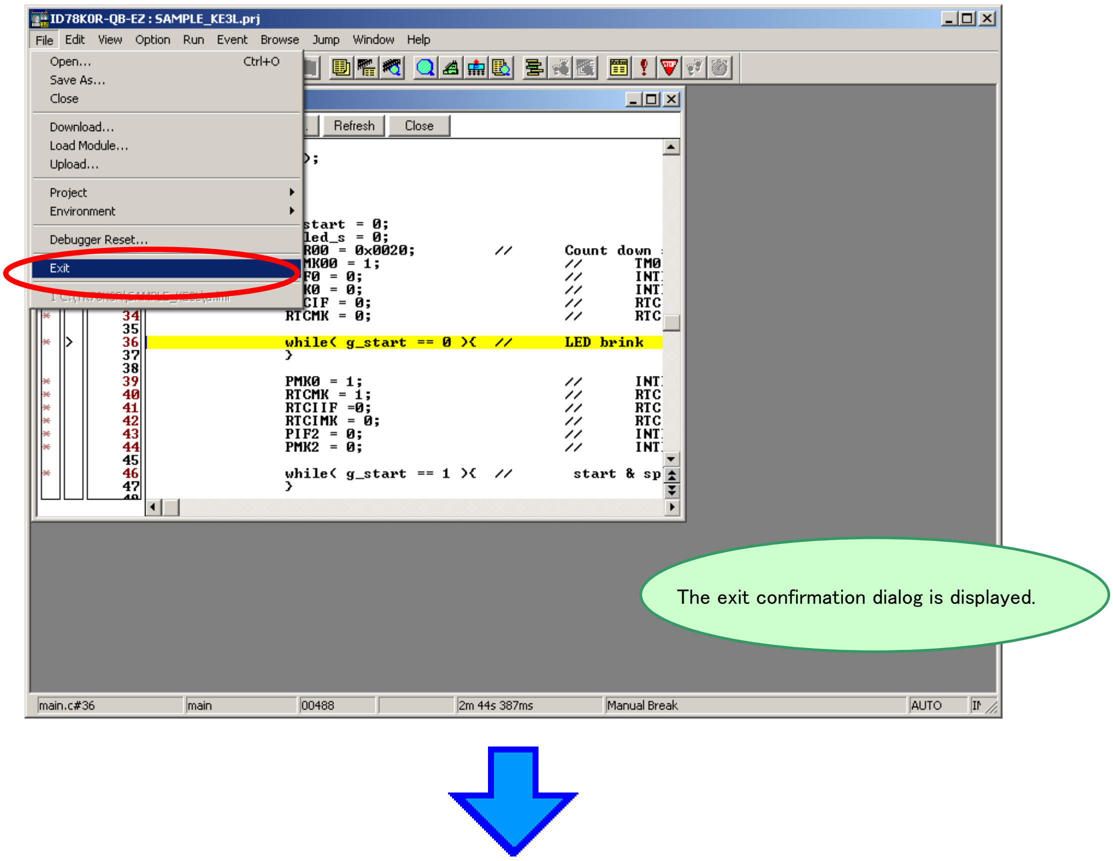


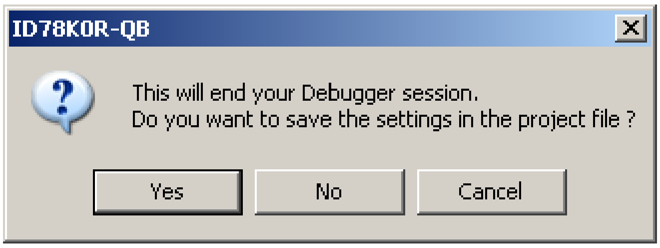


When the program stops, the status bar changes back to the original color.

2.12 Close Debugger (ID78K0R-QB-EZ)

Select "File" on menu bar, then "Exit".



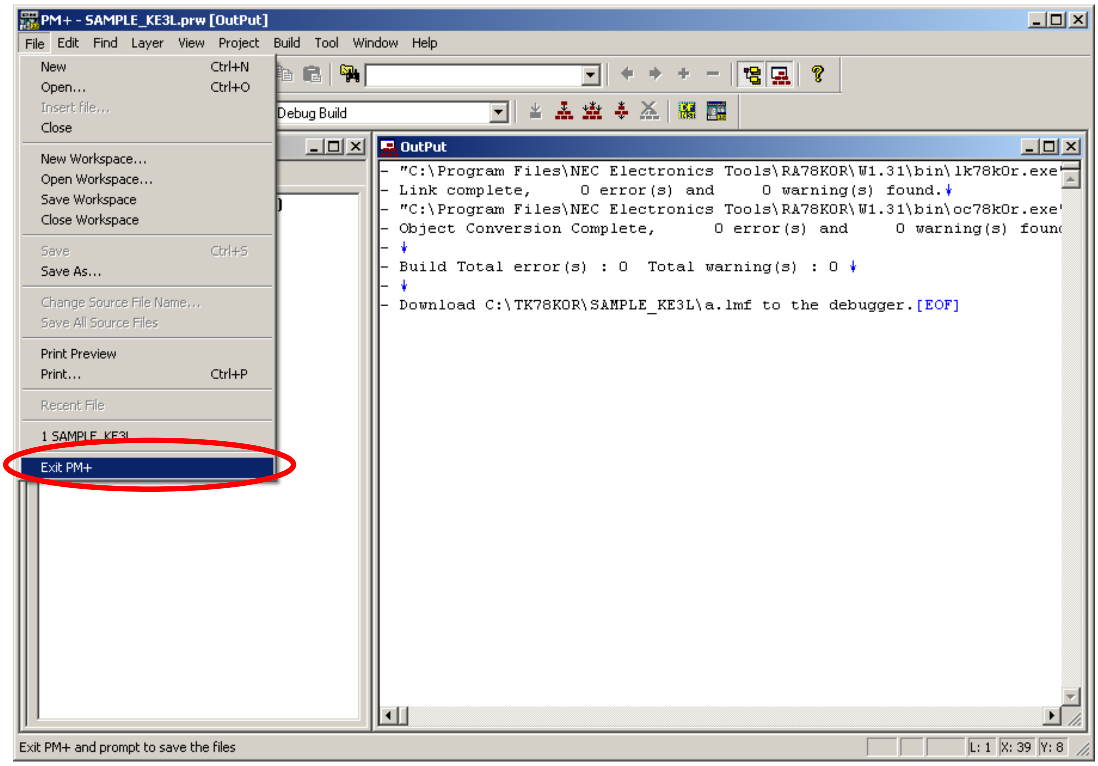


If you click [Yes], it saves the settings in the project file, and then closes the ID78K0R-QB-EZ. It is recommended to save the settings as it saves the window you used, window size, layout, etc.

If you click [No], it does not save the settings and closes the ID78K0R-QB-EZ.

2.13 Quit PM+

Select "File" on menu bar, then "Exit PM+".



PM+ is closed.

The experiences section ends now.
You can find more information how to use the development tool and information about other useful features in "Chapter 6 Other Information".

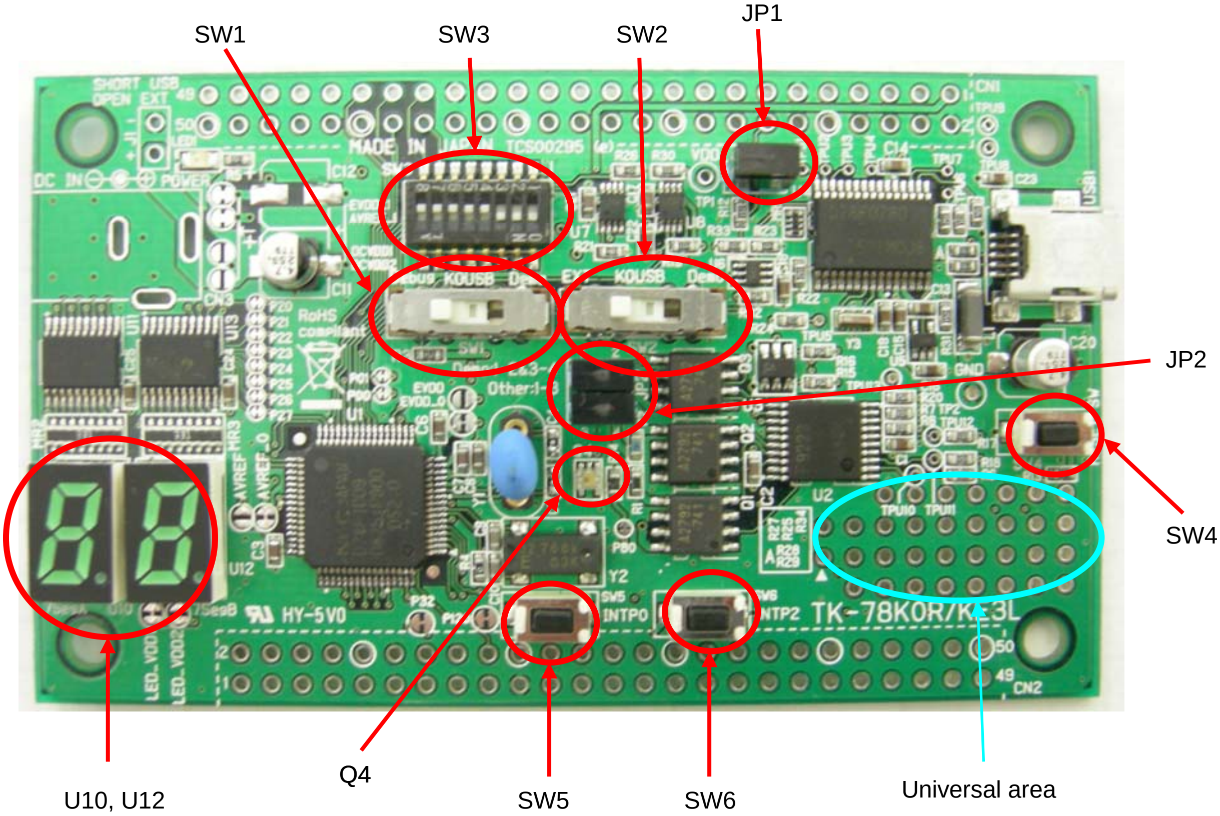
CHAPTER 3 Hardware Specifications

In this chapter, the hardware of TK-78K0R/KE3L will be explained.

Microcontroller	μPD78F1009 *78K0R/KE3-L
Clock	External main system clock: 20MHz Subsystem clock: 32.768KHz Internal high-speed oscillation : 1,8,20MHz
Interface	USB (USB1)
Power supply voltage	5V (USB or AC adapter)
Input/output for operation check use	・7segLED(U10,U12) ・Push switch (SW5,SW6) ・Dip switch (SW3) ・Universal Area ・Reset switch (SW4)
Other hardware	・Mode switch(SW1,SW2) ・Power LED(LED1)

* The name with bracket is the name printed on the board.

3.1 Layout of hardware functions



3.2 Hardware Functions

Following switch settings should be made depending on the mode.

Switch / Jumper No	Demo mode	Debug/Writing mode	PC communication mode
SW1	"Demo"	"Debug"	"K0USB"
SW2	"Demo"	"EXT"	"K0USB"
SW3-1～SW3-3	All OFF	All ON	All OFF
SW3-4～SW3-8	Not in use	Any	Any
JP1	Short	Short	Short
JP2	1-2, 3-4 Short	1-3 Short	1-3 Short

Demo mode　Set this when you use pre-installed low power consumption demonstration.

Debug/Writing mode　Set debug mode when you use bundled ID78K0R-QB-EZ. Set write mode when you use bundled WriteEZ4.

PC communication mode　Set this when you connect to K0/USB (uPD78F0730) UART6 that 78K0R/KE3-L UART1 is mounted on the board. As this is like connecting UART1 to COM port of PC, it can communicate with PC.

3.2.1 SW1

SW1 is a switch that has 3 positions: "Demo", "K0USB", and "Debug".

3.2.2 SW2

SW2 is a switch that has 3 positions: "Demo", "K0USB", and "EXT".

3.2.3 SW3

SW3 bit1～3 are for mode settings. Set with referring to the above table.
bit4～8 are DIP switches connected to ports in microcontroller.

Bit4～8 are connected to P33, P42, P43, P77, P76 pin in the microcontroller. Set this ON for "Low" and OFF for "Open". Before using this, you need to set the microcontroller built-in pull-up option resistor (PUx) to ON. (For details about settings for microcontroller built-in pull-up option resistor, refer to "78K0R/Kx3-L User's Manual <U19291J>".)

SW3

Bit 4	P33
Bit 5	P42
Bit 6	P43
Bit 7	P77
Bit 8	P76

3.2.4 SW4 (RESET SW)

SW4 is the reset switch. You can reset the microcontroller by pressing this switch.

3.2.5 SW5 (INTP1)

SW5 is the Push switch connected to "P120/INTP0/EXLVI" pin in microcontroller. When you push the switch, it becomes "Low", and when you release the switch, it becomes "Open". Before using this, you need to set the microcontroller built-in pull-up option resistor (PU12) to ON. (For details about settings for microcontroller built-in pull-up option resistor, refer to "78K0R/Kx3-L User's Manual <U19291J>".)

3.2.6 SW6 (INTP2)

SW6 is the Push switch connected to "P32/SCK10/SCL10/INTP2" pin in microcontroller. When you push the switch, it becomes "Low", and when you release the switch, it becomes "Open". Before using this, you need to set the microcontroller built-in pull-up option resistor (PU32) to ON. (For details about settings for microcontroller built-in pull-up option resistor, refer to "78K0R/Kx3-L User's Manual <U19291J>".)

3.2.7 JP1

JP1 is the jumper short pin to select power supply.

JP1

Short	Use USB power supply from USB1 connector
Open	Use external power supply from CN1 or CN2

3.2.8 JP2

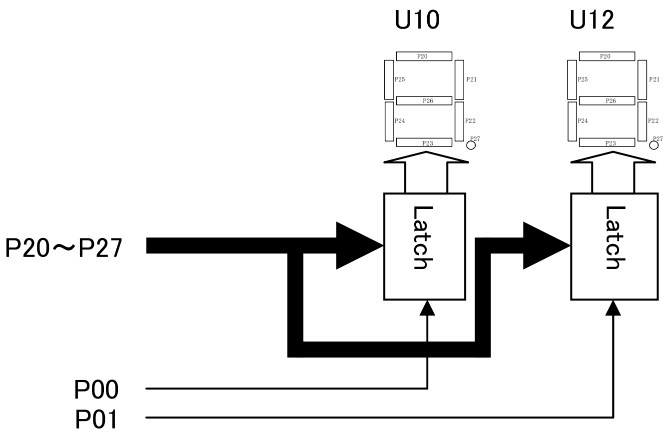
JP2 is the jumper for measure the power consumption of microcontroller.
When you use the demonstration, set it to 1-2, 3-4 short.
When you do not use the demonstration, set it to 1-3 short.

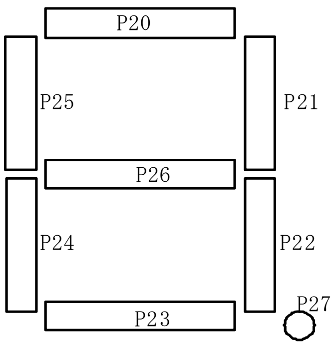
3.2.9 Q4 (Illuminance Sensor)

Q4 is an illuminance sensor that is connected to "P80/CMP0P/INTP3/PGAI" pin of the microcontroller.
When it gets lighter, the voltage increases, and when darker, the voltage decreases. Before using this, you need to set the microcontroller "P80/CMP0P/INTP3/PGAI" pin to programmable gain amplifiers input pin. (For details about register settings for microcontroller, refer to "78K0R/Kx3-L User's Manual <U19291J>".)

3.2.10 U10, U12 (7segLED)

U10 and U12 are 7segLED.
By setting the 7segLED output data in P20-P27 and setting P00 or P01 from Low to High, the data is latched and the 7segLED displays the data.





Example of data and its display

0	0xC0	5	0x92
1	0xF9	6	0x83
2	0xA4	7	0xf8
3	0xB0	8	0x80
4	0x99	9	0x98

To display "1" on U10 and "2" on U12:

```
PM2 = 0x00;                // Set P2 to output mode
PM0.0 = 0;  PM0.1 = 0;     // Set P00,P01 to output mode
P2 = 0xF9;                 // Set data for "1" in P2
P0.0 = 0; P0.0 = 1;        // Set P00 to Low, then High
P2 = 0xA4;                 // Set data for "2" in P2
P0.1 = 0; P0.1 = 1;        // Set P01 to Low, then High
```

3.2.11 LED1 (POWER)

LED1 is the POWER LED. It is lighted when it gets power supply.

3.3 Universal Area

The kit has the universal area. Users can use this to develop custom circuit.

CHAPTER 4 Troubleshooting

This chapter describes how to solve troubles you may face.

4.1 If you cannot find USB driver when you connect PC to the kit

Check Point 1

If you use USB hub, do not use it. (USB hub is not supported)

Check Point 2

Check if you installed "Starter Kit USB Driver" in "1.2 Installation of Development Tools". If not, install the driver.

Check Point 3

Check if the settings of SW5 on the kit are correct with referring to "1.3 Installation of USB Driver".

Check Point 4

If above 3 check points are confirmed, disconnect the USB cable from PC and re-connect again. It should show the "Found New Hardware Wizard" wizard. Operate the installation with referring to "1.3 Installation of USB Driver". After the installation, make sure you go through "1.3.3 Completion of USB Driver Installation" to confirm the USB driver installation.

4.2 Error when you start the debugger

There could be several reasons to make errors happen.
The solving processes differ depending on errors. Please check the error message first.
The solving processes for each error are as follows.

4.2.1 "Can not communicate with Emulator…" (F0100 or A0109)

Check Point 1

If you use USB hub, do not use it. (USB hub is not supported)

Check Point 2

Check if the settings of SW5 on the kit are correct with referring "1.3 Installation of USB Driver".

Check Point 3

Confirm the USB driver installation with referring to "1.3.3 Completion of USB Driver Installation".

Check Point 4

If above 3 check points are confirmed, close the debugger and disconnect the USB cable from PC. Re-connect USB cable properly to both the PC and the kit, and then re-start the debugger.

4.2.2 "Incorrect ID Code." (Ff603)

This error occurs when the security ID stored on microcontroller built-in flash memory is different from the ID code you entered at the start of debugger.

Security ID entry area at the start of debugger



Check Point 1

Enter correct security ID and click OK on the configuration window.

Check Point 2

If you forgot the security ID, you have to erase the microcontroller built-in flash memory. Before erasing, check if you actually set the security ID with referring to "2.4 Set Linker Options". Also remember the code you set for the security ID.
After this, erase the flash memory with referring to "5.4 Erase microcontroller built-in flash memory".

4.2.3 "The on-chip debug function had been disabled in the device." (F0c79)

This error occurs when the value at address C3H (On-chip debug option byte) in microcontroller built-in flash memory is incorrect. You need to erase the flash memory.

Check Point 1

Check if you actually set the correct on-chip debug option byte with referring to "2.4 Set Linker Options". If it is not correct, then set correctly.

Check Point 2

Erase the flash memory with referring to "5.4 Erase microcontroller built-in flash memory".

4.2.4 "Disabling the on-chip debug function is prohibited." (F0c33)

Basically, this error occurs when you start (download) the debugger without doing the settings described at "2.4 Set Linker Options". Do the same checking processes as "4.2.3 The on-chip debug function had been disabled in the device. (F0c79)".

CHAPTER 5 Other Information

This chapter explains some useful operation techniques of development tools and circuit diagram of the kit for developing of user programs.

5.1 Create a new workspace (project)

5.2 Register additional source file

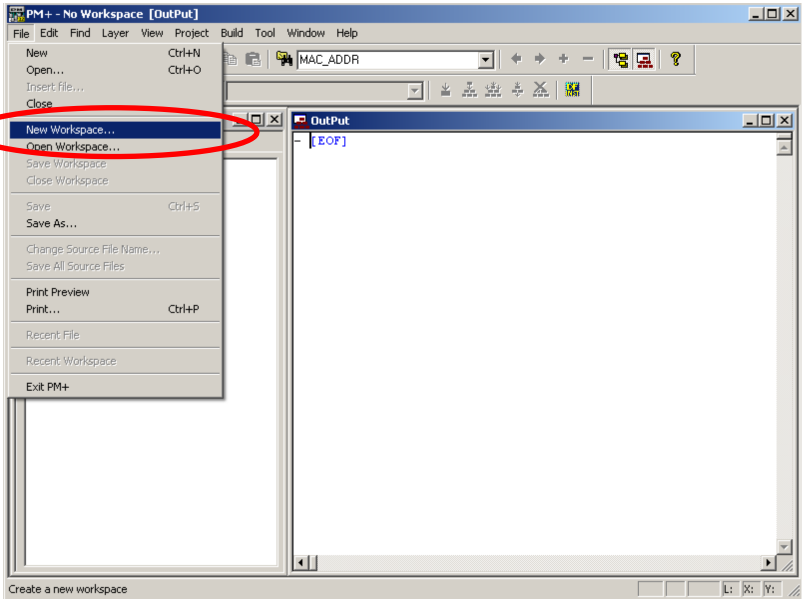
6.3 Debugger tips

6.4 Erase microcontroller built-in flash memory

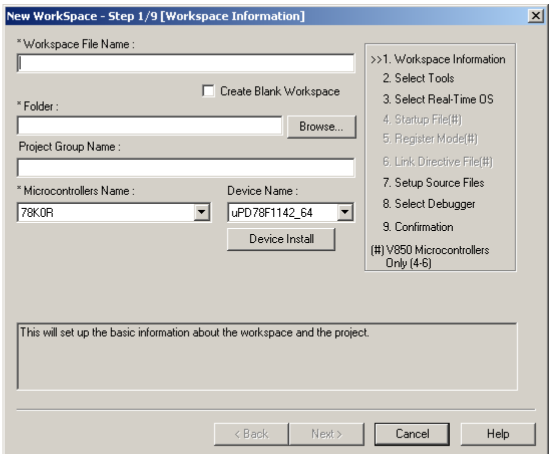
6.5 Circuit diagram

5.1 Create a new workspace

Now, create a new workspace and project.
PM+ allows you to create a new workspace with following "New WorkSpace" dialog.
Select "File" on PM+ menu bar, then "New Workspace...".



"New WorkSpace" dialog opens



The concrete information set here is described on the following pages

<Description of items>

Workspace File Name:
-> Specify the name of the workspace file that manages the project files.
.prw is automatically suffixed as the file type.
A project file (.prj) of the same name is simultaneously created.

Folder:
-> Specify the folder for saving the workspace file by writing its absolute path.
This item can be selected from a reference dialog box by pressing the Browse... button.

Project Group Name:
-> Specify this item if wishing to manage multiple projects together in function units.
If nothing is specified, this item is the same as the workspace file name.

Microcontroller Name:
-> Specify the name of the microcontroller to be used.

Device Name:
-> Specify the name of the device to be used.

Input the workspace information setting as follows.

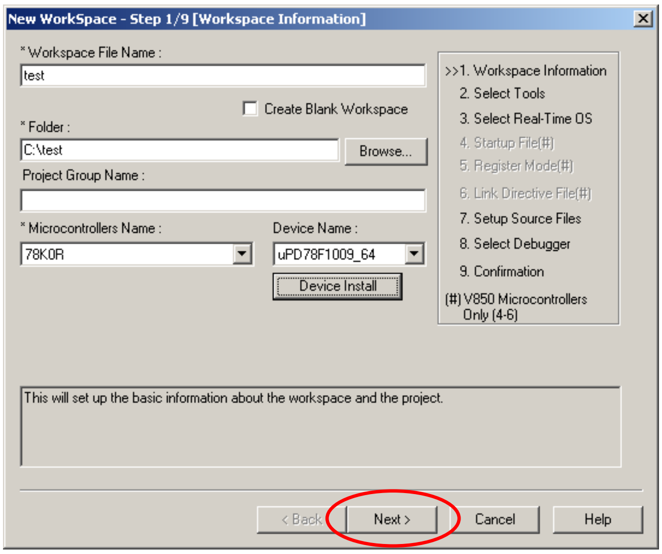
Workspace file name
→ test

Folder
→ C:¥test

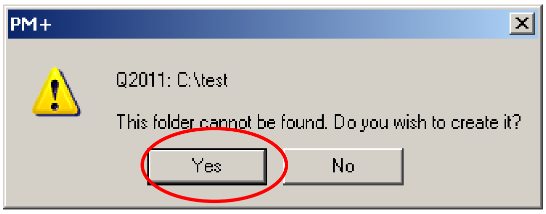
Project Group Name
→ (no input)

Microcontroller Name
→ 78K0R

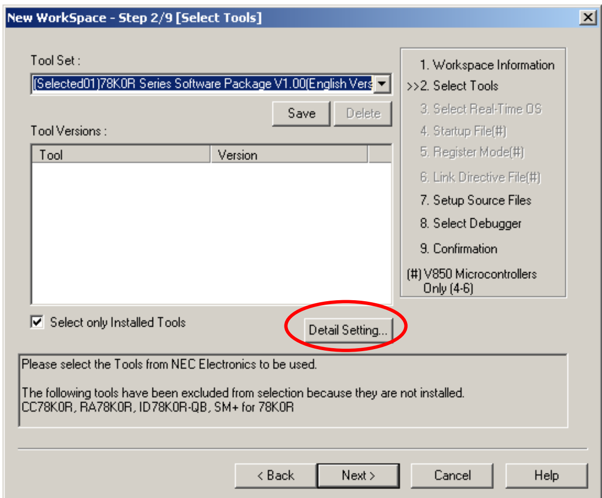
Device Name
→ uPD78F1009_64



Click Next > button

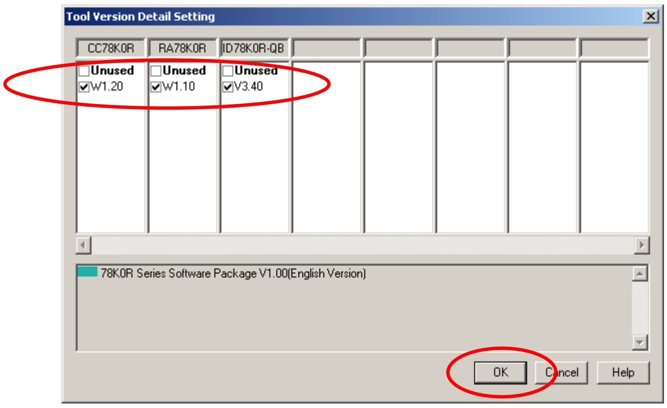


Click Yes button

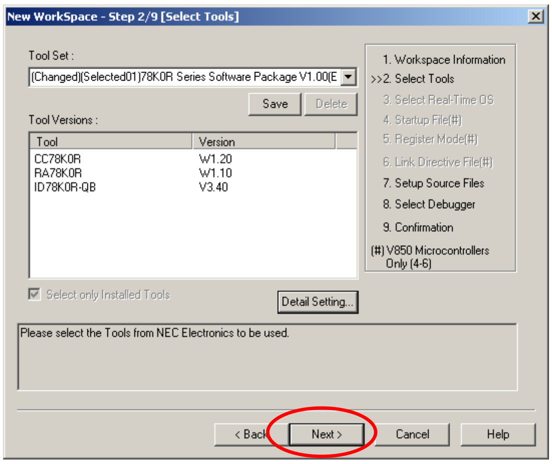


Click Detail Setting button

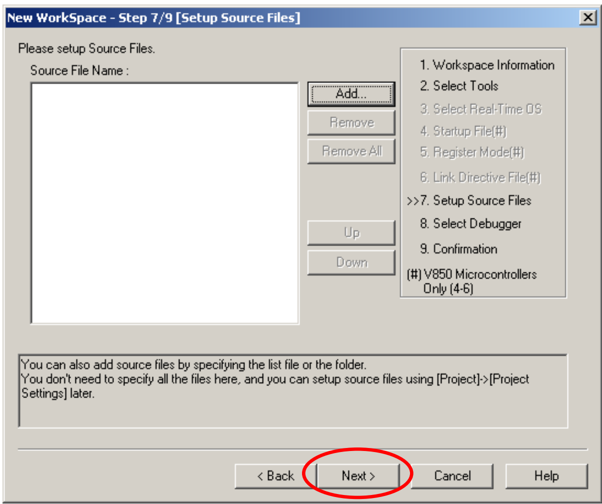
Set the version of tools as follows.
CC78K0R:W2.10
RA78K0R:W1.31
ID78K0R-QB-EZ:V3.50



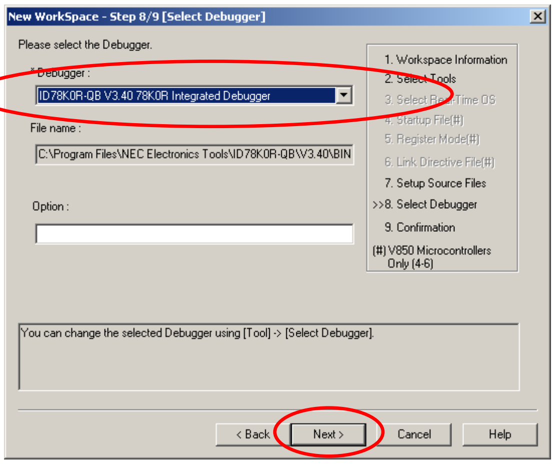
Select tools as above screenshot, then click OK .



Click Next >



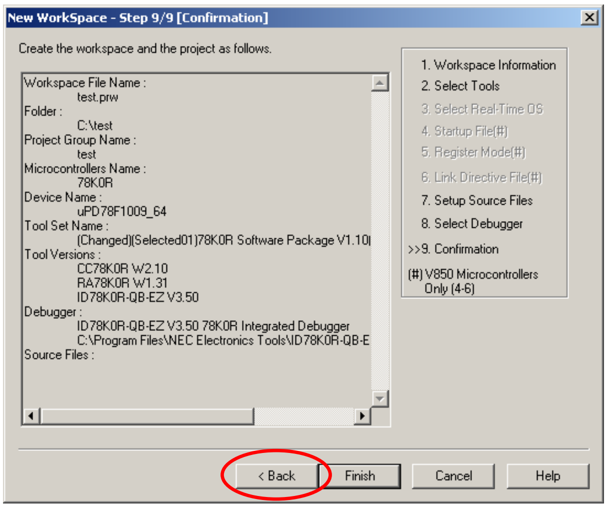
Click Next >



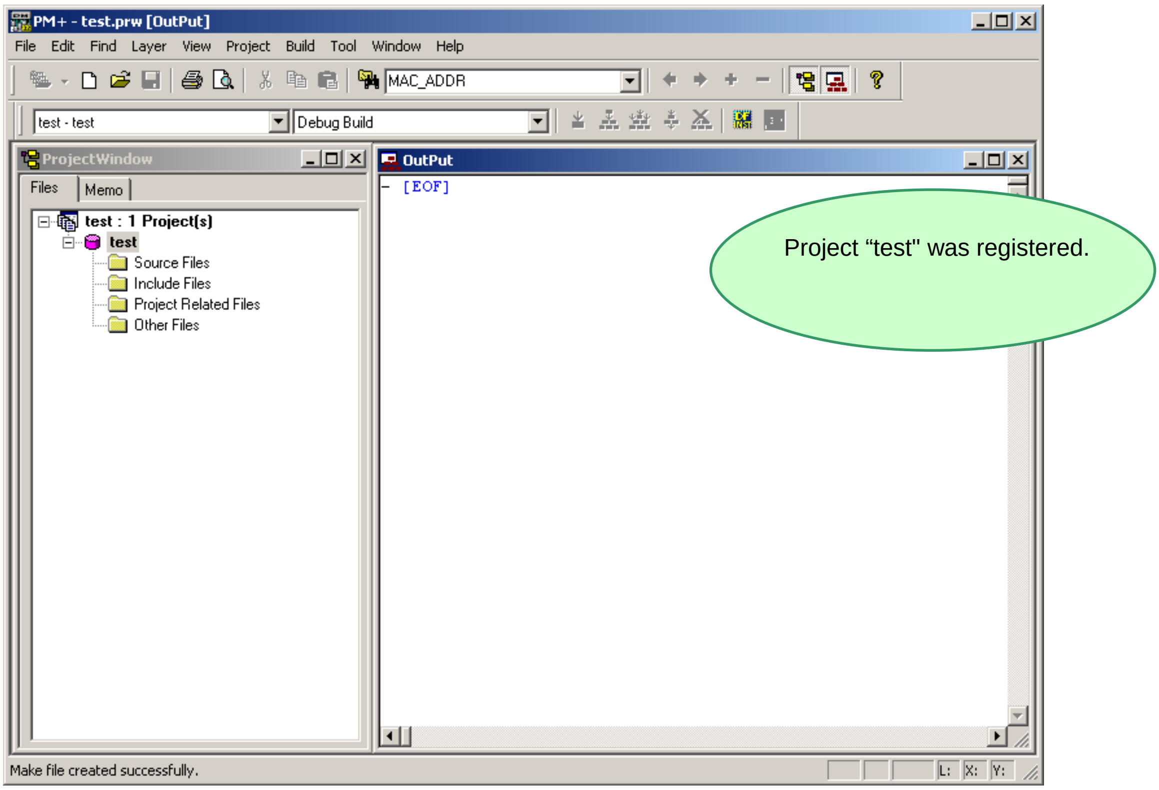
Select ID78K0R-QB-EZ V3.50

Click Next >

Check the project information settings



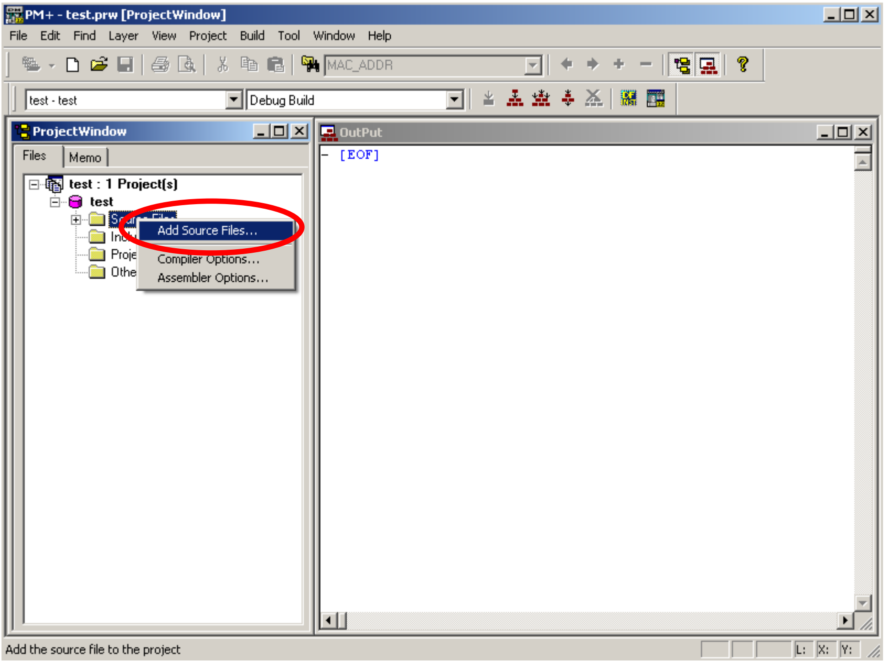
Click Finish

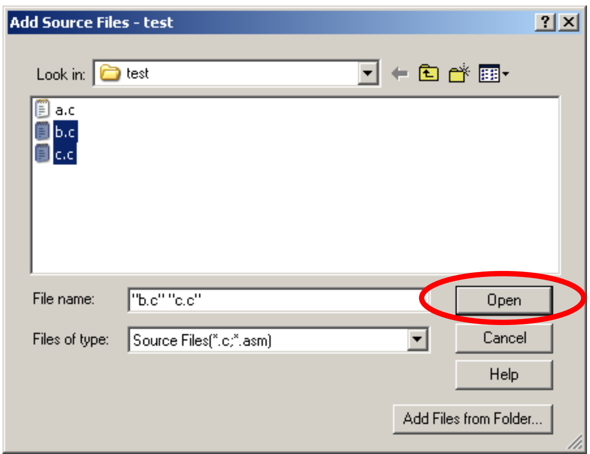


This completes workspace and project creation.
Additional source files can be registered at any time thereafter.
For details, refer to "6.2 Register additional source file".
Also, you need to do the settings for on-chip debug. Please refer to "2.4 Set Linker Options", "2.5 Set Compiler Options", and "2.7 Check Debugger Settings".

5.2 Register additional source file

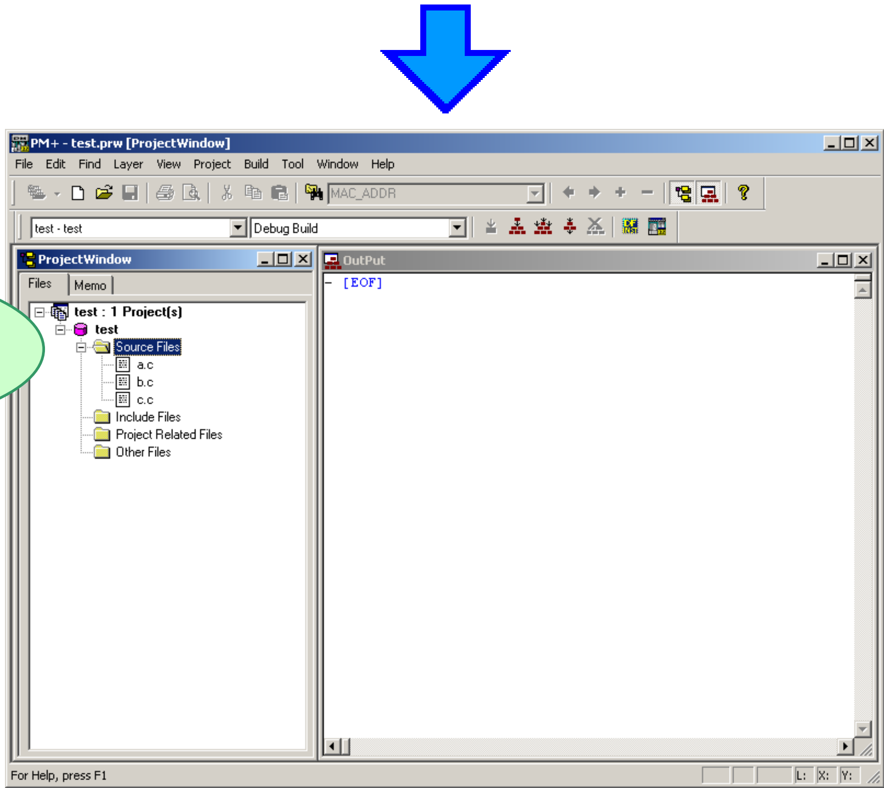
Now, register additional source files.
The following example shows the additional registration of source files "b.c" and "c.c" with source file "a.c" already registered.
Place the cursor on the source file in the Project window of PM+, and select [Add Source Files…] displayed in the right-click menu.





Select source files "b.c" and "c.c", then click Open

Multiple source files can be selected by clicking them with pressing Ctrl key.



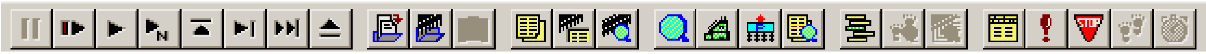
Source file "b.c" and "c.c" are additionally registered to the project.

5.3 Debugger tips

This section describes some useful techniques for the debugger (ID78K0R-QB-EZ).

5.3.1 Change display of buttons

Execution controls (run, stop, step-in debugging, reset, etc) and opening functional window can be made by below buttons. However, it could be difficult to know which button does what.



In this case, select "Options" on menu bar, then "Debugger Options". Check "Pictures and Text" on setting area.

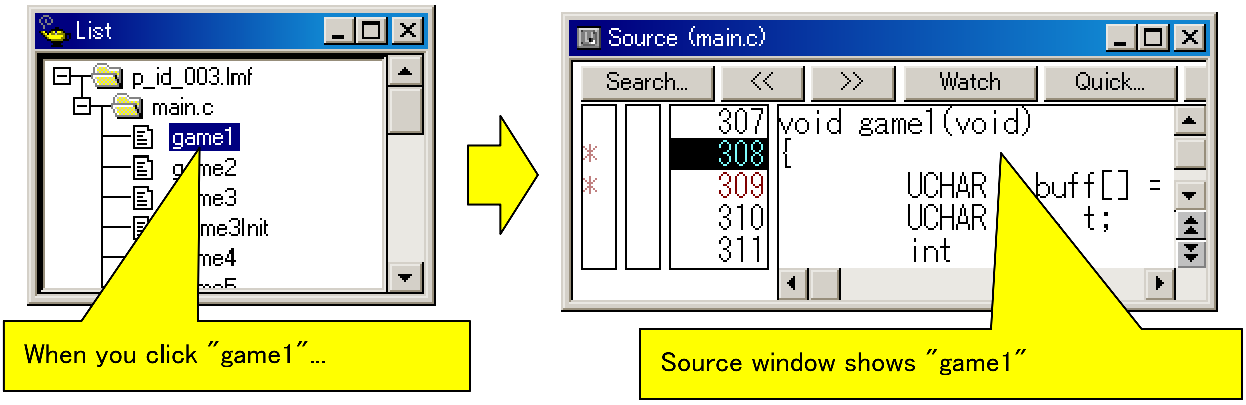


With this setting, the buttons display the text as well, so that it is easier to know what they are.



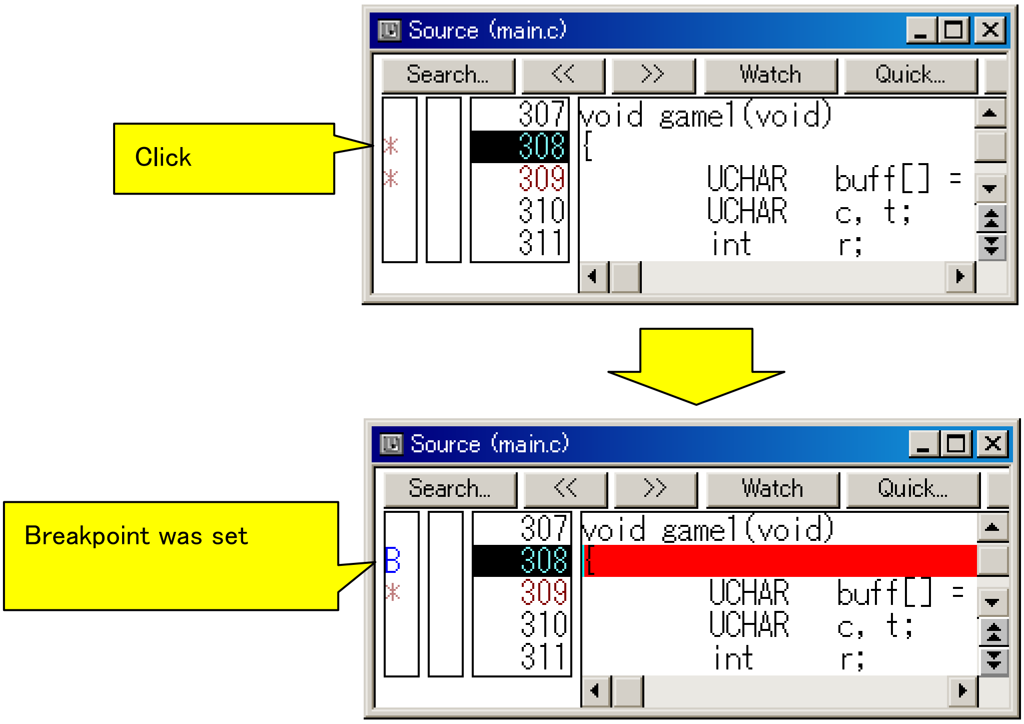
5.3.2 Display source list and function list

When you wish to see source file list or function list, select "Browse" on menu bar, then "Other" -> "List" to open the list window. The information in the windows is synchronized. Therefore, it is not just for referring to the list, but it is useful when you wish to update files or functions.



5.3.3 Set/delete breakpoints

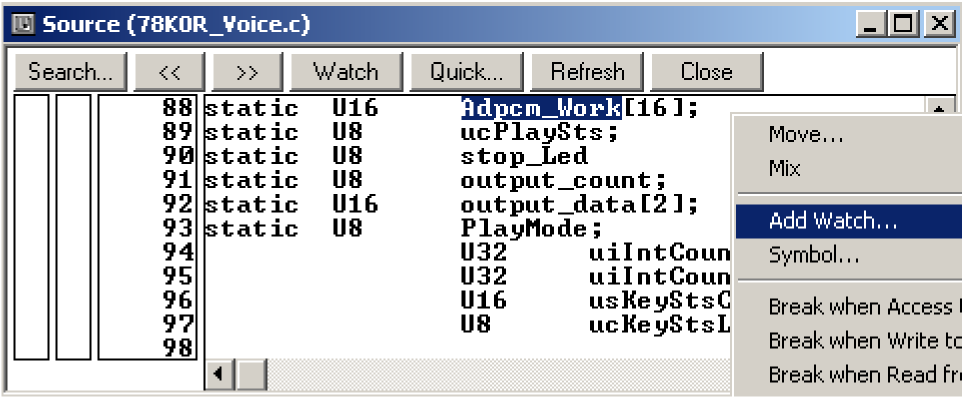
Breakpoints are executed by clicking lines in which " * " is displayed
"B" is displayed in the line where a breakpoint is set.
Breakpoints are deleted by clicking "B".



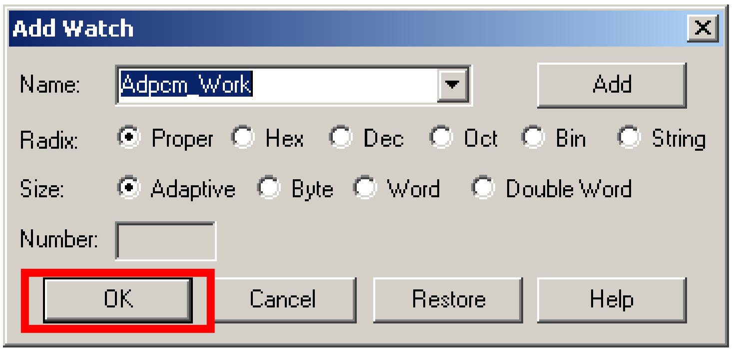
5.3.4 Display global variables

With using Watch Window, you can display global variables. There are several ways to register global variables to watch window. In this section, how to register from source window is described.

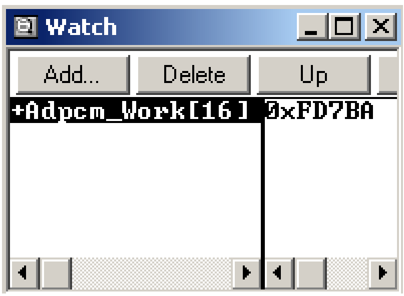
①Right-click the variable on source window, then select "Add Watch…"



②Add Watch dialog opens. Click OK .



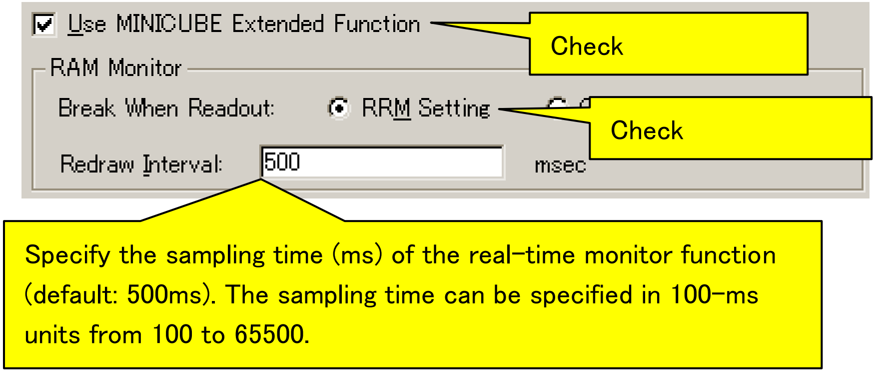
③Adding a variable to watch window is completed.



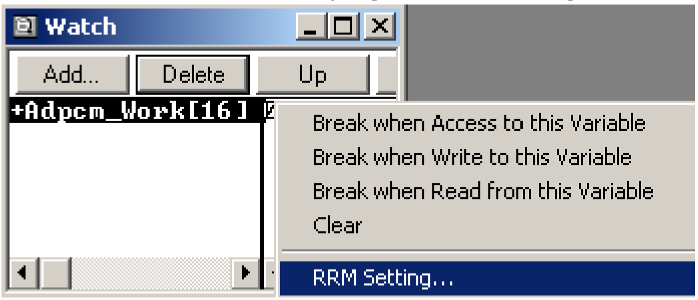
5.3.5 Display global variables while programs are running

Global variable could be referred even when the programs are running.

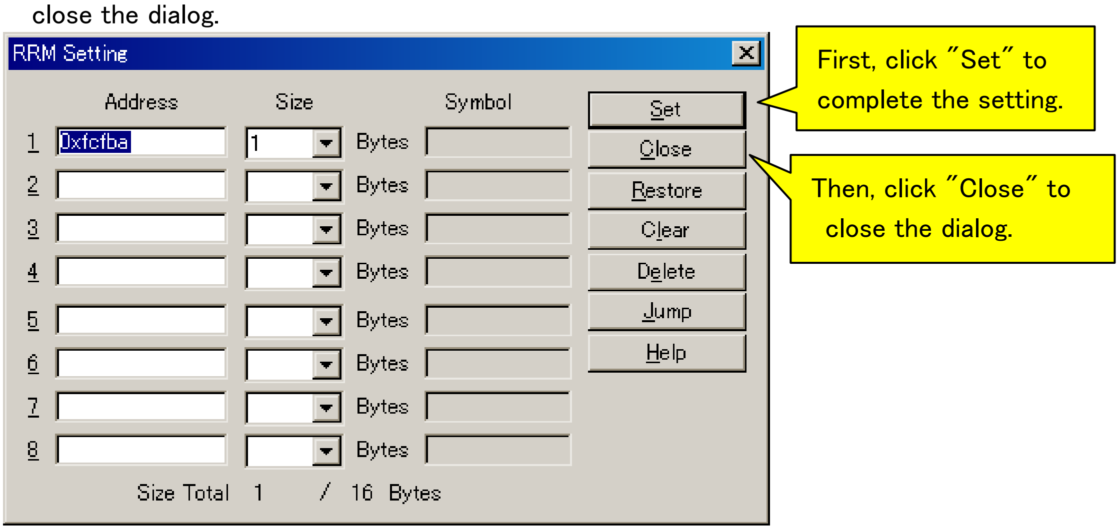
①Select "Option" menu -> "Extended Option...". Follow below settings.



②Right-click the variable in watch window while programs are running, then select "RRM Setting...".



③RRM Setting dialog opens. Click Set button to complete the setting, then Close button to close the dialog.



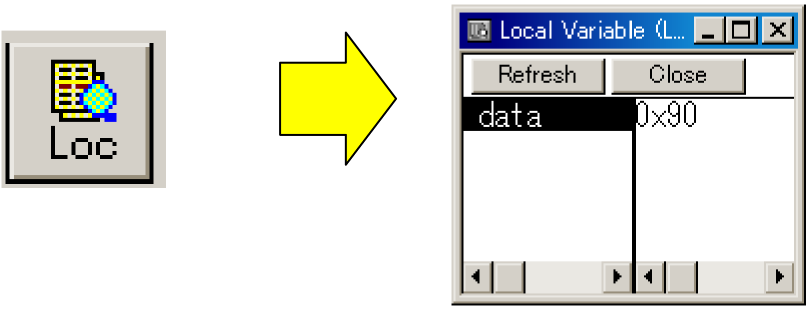
This completes the settings.

CAUTION:

- The maximum size of variable area is as total of 16byte when programs are running.
- The maximum number of variable area is 8 locations
- The user program momentarily breaks upon a read.

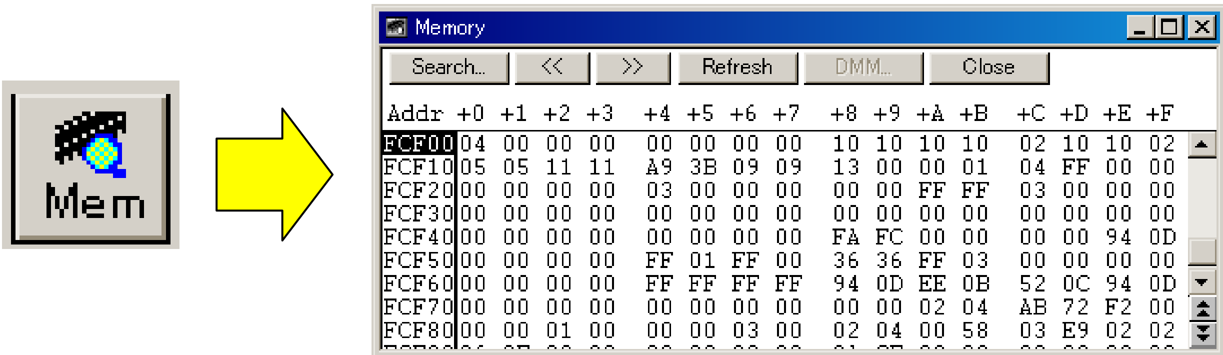
5.3.6 Display local variables

Local variable window is used to display local variables.
By clicking the button below, you can open the local variable window.
Unlike global variables, local variables cannot be displayed when programs are running.

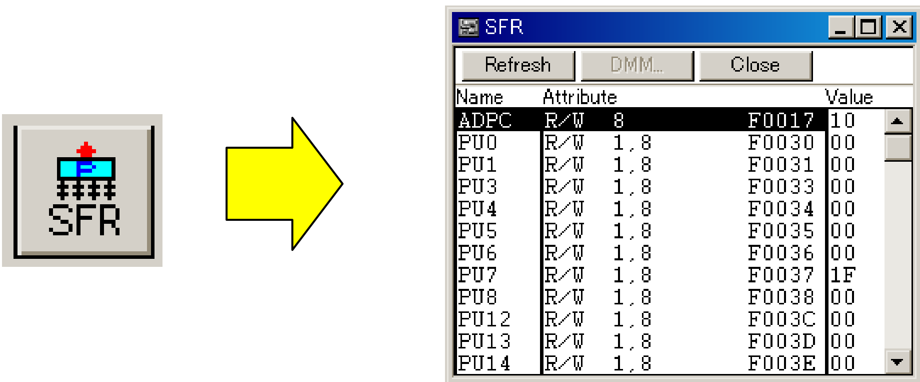


5.3.7 Display memory and SFR contents

By clicking the button below, you can open the memory window.



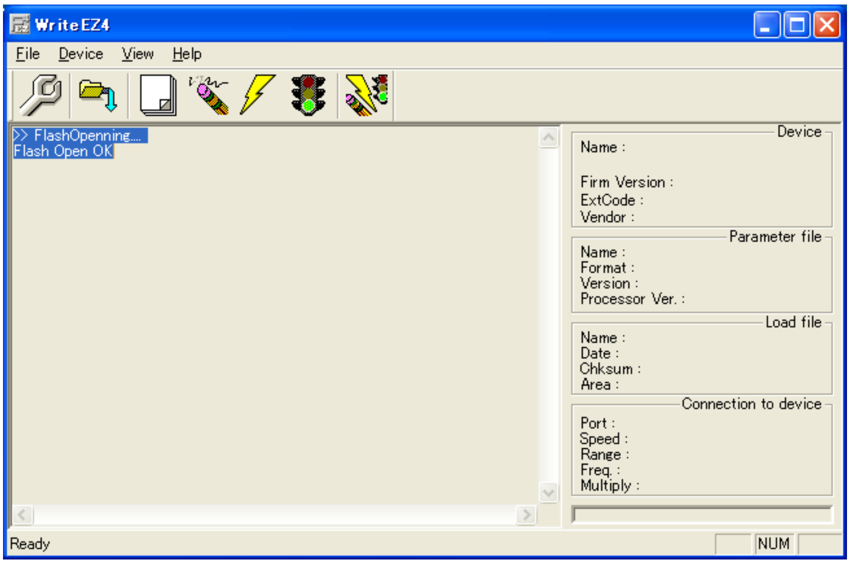
By clicking the button below, you can open the SFR window.



5.4 Erase microcontroller built-in flash memory

WriteEZ4 can erase the flash memory, when you forgot the security ID or you set unexpected value. By erasing the flash memory, you can reset the security ID to "FFFFFFFFFFFFFFFFFFFF".

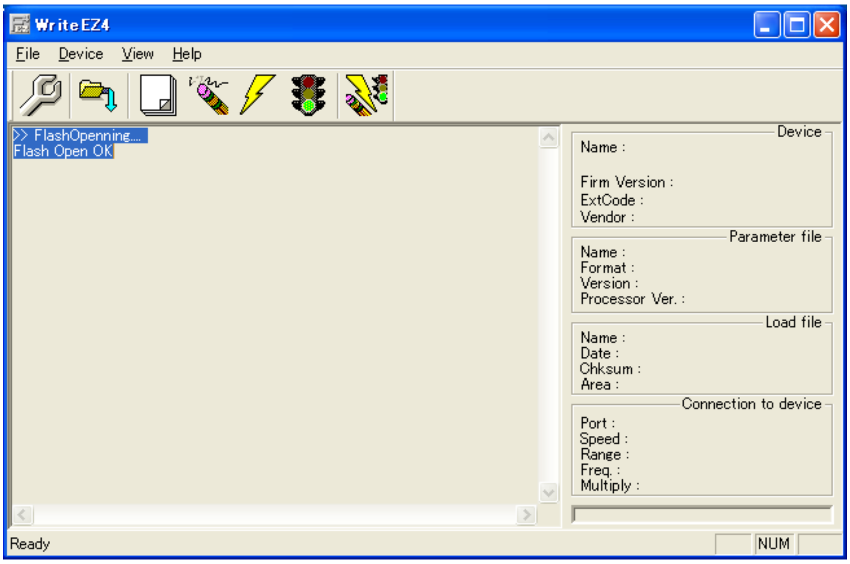
Start WriteEZ4 from NEC Electronics Tools.



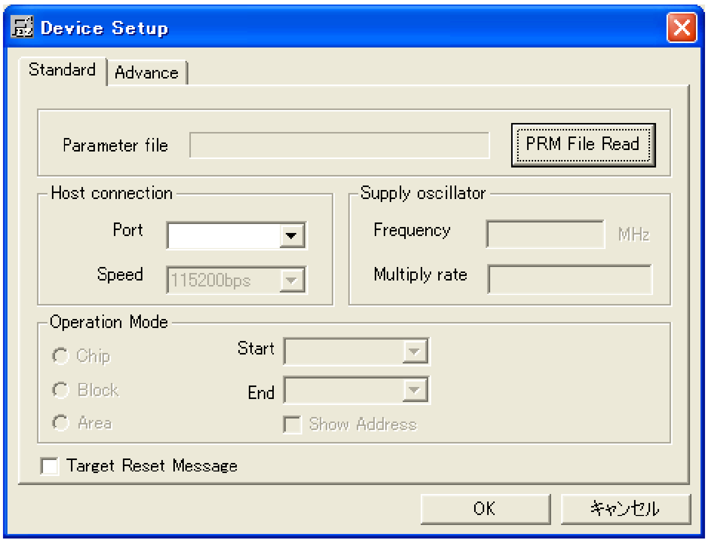
①Set TK-78K0R/KE3L switches to "Debug/Writing mode", and then connect the PC.

Switch / Jumper No	Demo mode	Debug/Writing mode	PC communication mode
SW1	"Demo"	"Debug"	"K0USB"
SW2	"Demo"	"EXT"	"K0USB"
SW3-1～SW3-3	All OFF	All ON	All OFF
SW3-4～SW3-8	Not in use	Any	Any
JP1	Short	Short	Short
JP2	1-2, 3-4 Short	1-3 Short	1-3 Short

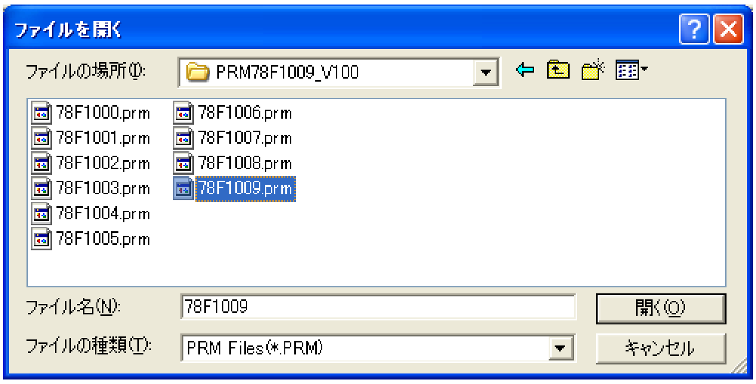
②Click the setup button.



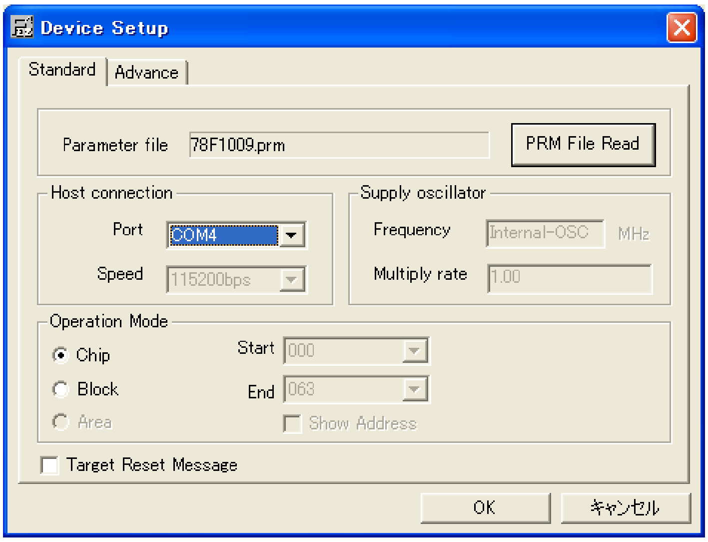
③Click "PRM File Read" button.



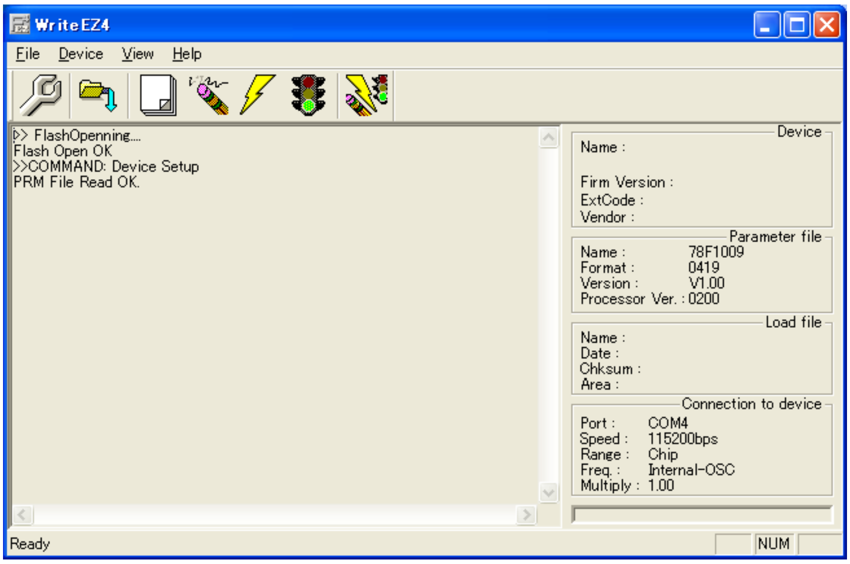
③Select "78F1009.prm" under the directory "C:¥Program Files¥NEC Electronics Tools¥WriteEZ4¥V1.02¥WriteEZ4¥PRM78F1009_V100".



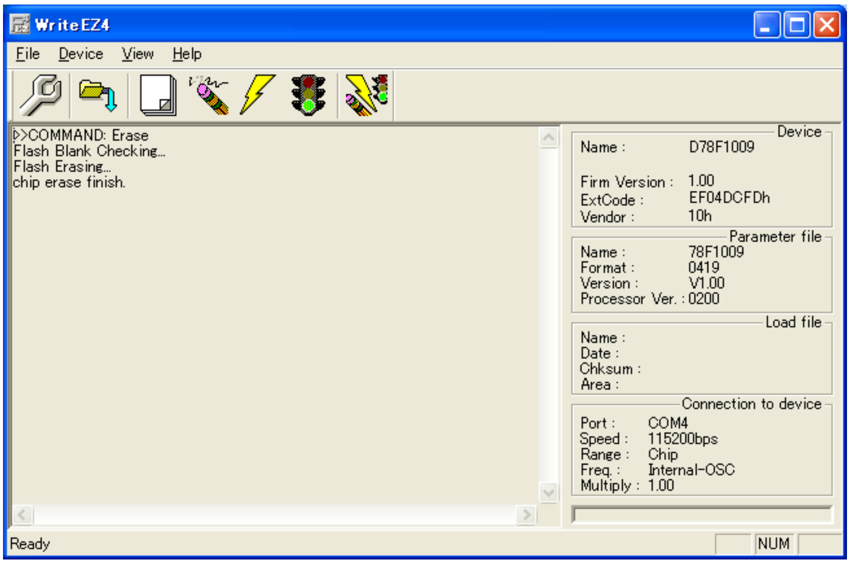
④Select the COM port that TK-78K0R/KE3L is connected.



⑤Click "Erase" button.

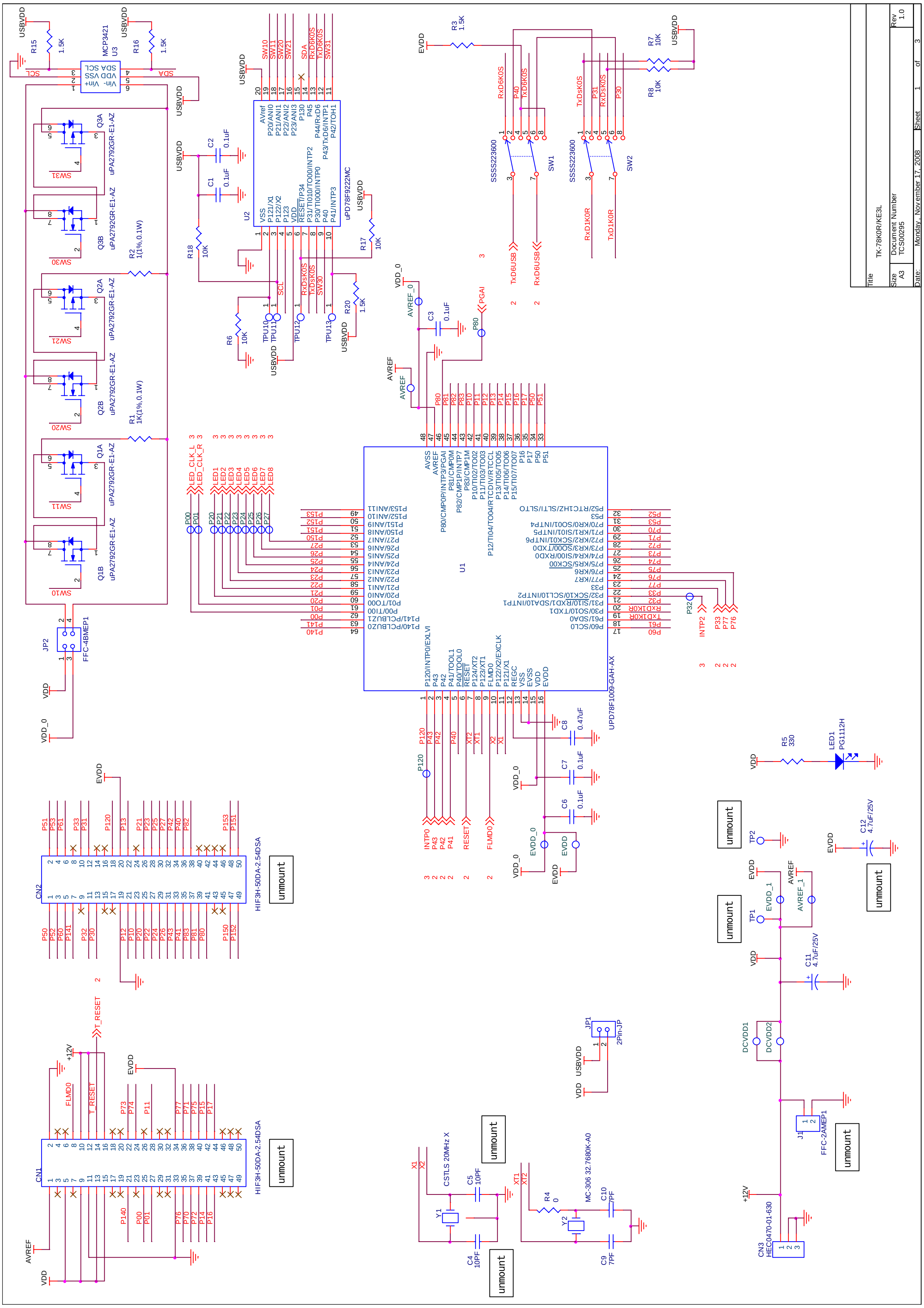


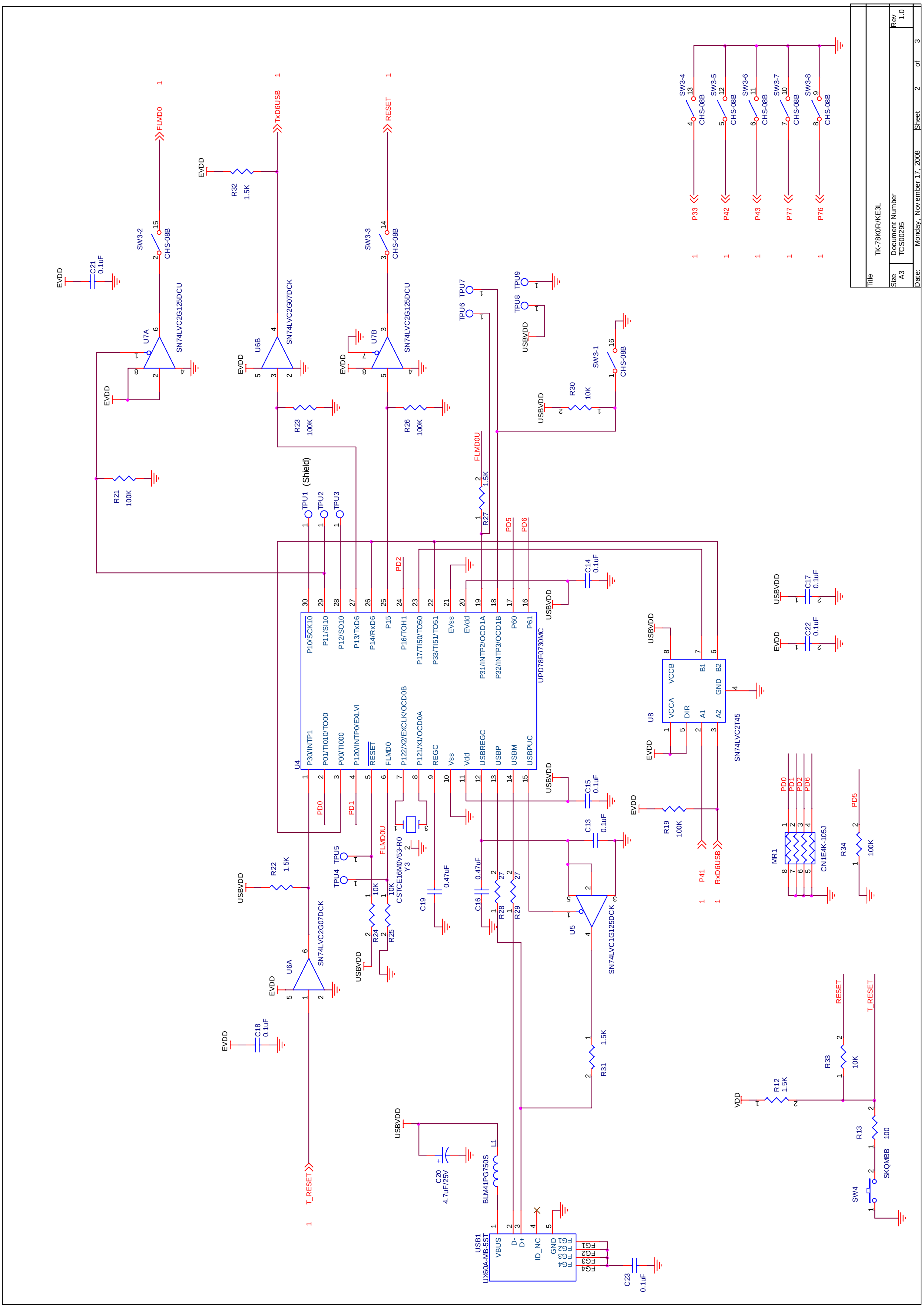
⑥Erasing the flash memory is completed when "chip erase finish" is displayed like below screenshot.

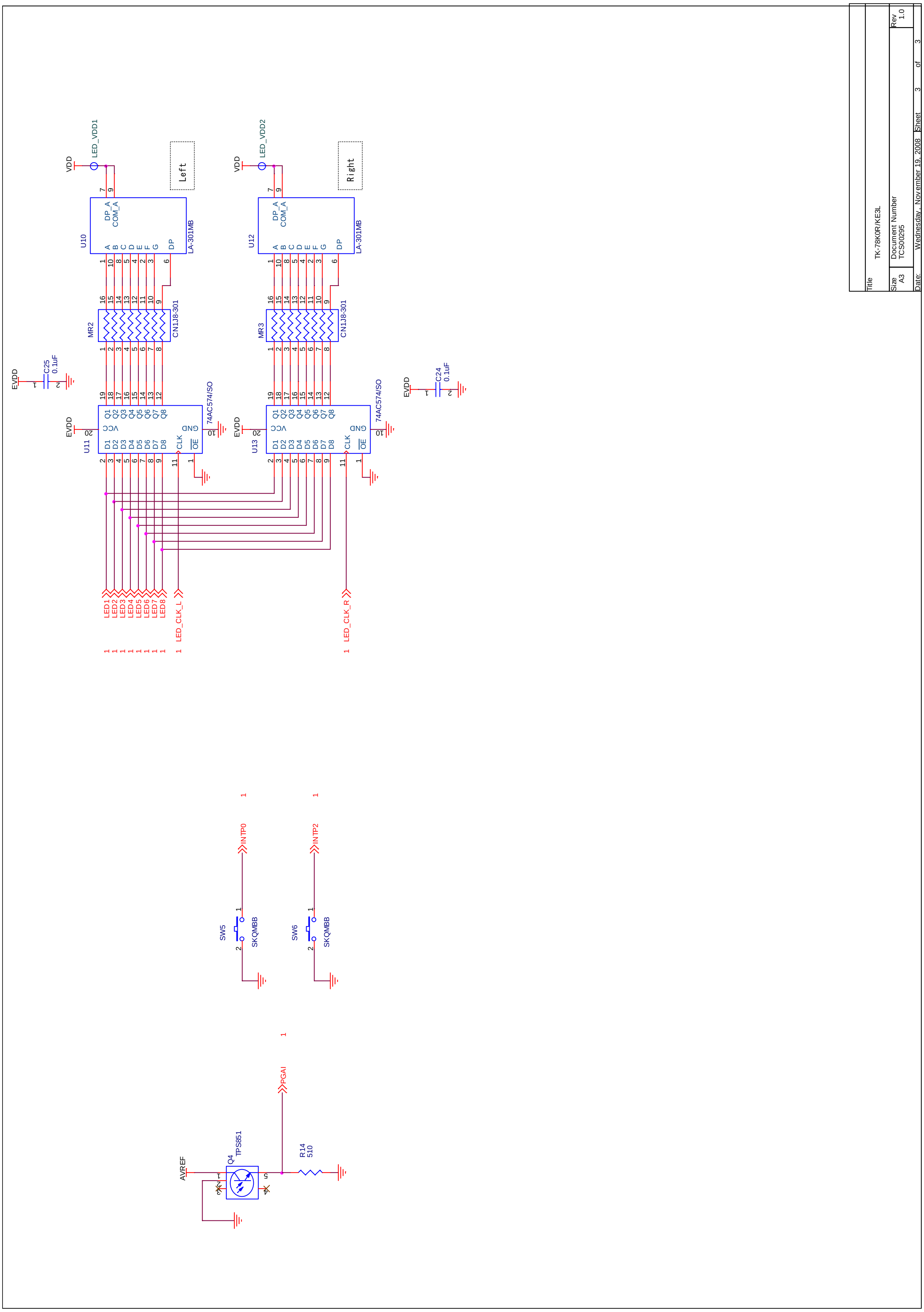


5.5 Circuit diagram

From following page, it shows the circuit diagram of the demonstration kit.


Title
TK-78K0R/KE3L
Size
A3
Document Number
TCS00295
Rev
1.0
Date: Monday, November 17, 2008
Sheet 1 of 3


Title
TK-78K0R/KE3L
Size
A3
Document Number
TCS00295
Rev
1.0
Date: Monday, November 17, 2008
Sheet 2 of 3


Title
TK-78K0R/KE3L
Size
A3
Document Number
TCS00295
Rev
1.0
Date: Wednesday, November 19, 2008
Sheet 3 of 3
U10
LA-301MB
U12
LA-301MB
Left
Right
LED_VDD1
LED_VDD2
VDD
MR2
CN1J8-301
MR3
CN1J8-301
U11
74AC574/SO
U13
74AC574/SO
EVDD
C25
0.1uF
C24
0.1uF
LED1
LED2
LED3
LED4
LED5
LED6
LED7
LED8
LED_CLK_L
LED_CLK_R
SW5
SKQMBB
SW6
SKQMBB
INTP0
INTP2
Q4
TPS851
AVREF
R14
510
PGAI